



UniCEUB – Centro Universitário de Brasília

FATECS – Faculdade de Tecnologia e Ciências Sociais Aplicadas

Curso de Engenharia da Computação

Projeto Final

SISTEMA DE MONITORAMENTO POR UM FEIXE DE LUZ

Por

Luiz Henrique Santiago Madeira campos

RA: 20064654

Professor Orientador:

Prof. MC. Claudio Penedo

Brasília, DF – Julho 2008.



UniCEUB – Centro Universitário de Brasília

FATECS – Faculdade de Tecnologia e Ciências Sociais Aplicadas

Curso de Engenharia da Computação

Projeto Final

SISTEMA DE MONITORAMENTO POR UM FEIXE DE LUZ

Monografia apresentada à banca examinadora para conclusão do curso e obtenção do título de bacharel em Engenharia da Computação do Centro Universitário de Brasília - UniCEUB

Brasília, DF – Junho 2008.

AGRADECIMENTOS

Em primeiro lugar, agradeço a Deus por me dar saúde e força para conquistar mais uma etapa da minha vida.

Ao meu pai Leonel Francisco Barbosa Madeira Campos, minha mãe Maria de Fátima Cartucho Madeira Campos, e meus irmãos, por me apoiarem em todos os momentos.

A todos os colegas de faculdade pelo apoio e incentivo. Em especial ao Bruno Márcio, Diogo e Marcito Ribeiro Madeira Campos, por passar seus conhecimentos e sabedorias durante essa caminhada.

A minha namorada Jaqueline Moura Machado e amigos que sempre acreditaram em mim e entenderam todos os momentos em que estive ausente, vocês também fazem parte dessa conquista!

Ao meu professor orientador MC. Claudio Penedo, pelo apoio.

E a todo o corpo docente da instituição UniCEUB, meu muito obrigado.

RESUMO

A escolha desse projeto foi influenciada por perceber que hoje em dia diversas empresas vêm se prevenindo com relação à segurança de suas empresas e residências, não medindo esforços para que tenha um bom sistema de segurança. Porém, nem todo mundo tem condições de adquirir um sistema sofisticado para colocar em seu estabelecimento. Isto motivou a elaboração de um projeto que seja eficiente na monitoração de estabelecimentos e possua um baixo custo.

Neste projeto é abordada a construção de um protótipo eletrônico, e software de detecção a partir de um feixe de luz infravermelho, capaz de filmar qualquer objeto que o interrompa.

No hardware desenvolvido são utilizados um sensor foto transistor onde é focalizado o feixe de luz, um buzzer responsável por soar um alarme quando houver alguma interrupção no feixe de luz, um conversor serial para comunicação com o computador, uma câmera responsável por filmar qualquer intervenção no feixe luminoso e juntamente com todos estes dispositivos, usa-se o microcontrolador da Atmel.

O software foi desenvolvido utilizando a linguagem de programação assembly e visual basic responsável por monitorar a filmagem que está sendo feita e armazenar no computador o vídeo que foi gravado.

Palavra-Chave: feixe de luz, linguagem de programação assembly e visual basic, buzzer, conversor serial e Microcontrolador.

ABSTRACT

The choice of this project was influenced by realizing that nowadays many companies has been preventing about the security of its enterprises and households, not measuring efforts to have a good security system. But not everyone is able to acquire a sophisticated system to put in their establishment, with that drafted a project that has a high level of security while not had a cost so high.

This project deals with the construction of a prototype electronic, and software for detecting from a beam of infrared light, capable of shooting any object that the stop.

In Hardware developed a sensor is used photo transistor which is focused the beam of light, a buzzer responsible for sound an alarm when there is any interruption in the beam of light, a converter for serial communication with the computer, a camera responsible for any intervention in shooting light beam and together with all devices, is used, the Atmel microcontroller.

The software was developed using the assembly programming language and visual basic responsible for monitoring the filming being done on the computer and store the video that was recorded.

Keyword: beam of light, the assembly programming language and visual basic, buzzer, serial converter and Microcontroller.

SUMÁRIO

CAPITULO 1 – INTRODUÇÃO	1
1.1 Estrutura do Trabalho	2
CAPITULO 2 – FUNDAMENTAÇÕES TEÓRICAS.....	3
2.1 Sistemas de Monitoramento	3
2.2 Luz Infravermelho	5
CAPITULO 3 – REFERÊNCIAL TECNOLÓGICO	7
3.1 Sensores	7
3.1.1 - Sensor Magnético ou reed switch.....	7
3.1.2 - Reed switch	8
3.1.3 - Sensor Infravermelho Passivo.....	8
3.1.4 - Infravermelho Ativo ou Feixe (IVA)	9
3.1.5 - Microondas (MO).....	10
3.2 Modelos de Webcam.....	110
3.3 Placa de captura de vídeo.....	11
CAPITULO 4 – DESCRIÇÃO DO HARDWARE	13
4.1 Apresentação Circuito Eletrônico	15
4.2 Características do Microcontrolador Atmel AT89C2051	16
4.3 Interface Serial	17
4.3.1 - Formato dos Dados Transmitidos.....	18
4.3.2 - Comunicação Serial do Projeto	18
4.4 Câmera Utilizada para Capturar a imagem	19
4.5 Dispositivo Infravermelho	20
4.6 Protótipo Desenvolvido	21
CAPITULO 5 – DESCRIÇÃO DO SOFTWARE	23
5.1 Fluxo de Comunicação.....	24
5.1.2 - Fluxo de Comunicação Serial.....	25
5.2 VideoCapX	26
5.3 Captura de Imagem.....	27
CAPÍTULO 6 - CONCLUSÃO	30
6.1 Propostas Futuras	30
REFERÊNCIAS BIBLIOGRÁFICAS	31
Apêndice 1 – Configuração do Microcontrolador	32

Apêndice 2 – Configuração do VideoCapx	39
--	----

LISTA DE FIGURAS

Figura 2.1 - Modelo exemplo de sistema de monitoramento.	3
Figura 2.2 - A luz infravermelha constitui uma pequena parte do espectro luminoso.	5
Figura 3.1 - Sensor Infravermelho Passivo.	8
Figura 3.2 - Tipos de Webcam.	10
Figura 3.3 - Saídas de áudio e vídeo composto /Conexão USB.	11
Figura 3.4 - Conector e portas USB.	11
Figura 3.5 - Placa de Captura.	12
Figura 4.1 - Esquemático do Módulo de Detecção.	13
Figura 4.2 - Circuito Desenvolvido.	14
Figura 4.3 - Layout do Circuito.	15
Figura 4.4 - Microcontrolador Atmel AT89C2051.	16
Figura 4.5 - Conversor Serial USB.	17
Figura 4.6 - Transmissão de uma Porta Serial.	18
Figura 4.7 - Configuração do Conector Serial.	19
Figura 4.8 - Câmera kemex AW-1035.	19
Figura 4.9 - Dispositivo que Emite Luz Infravermelho.	20
Figura 4.10 - Protótipo desenvolvido.	21
Figura 5.1 - Fluxo da Comunicação.	24
Figura 5.2 - Fluxo de Comunicação Serial.	25
Figura 5.3 - Tela Inicial de Instalação do VideoCapX.	27
Figura 5.4 - Fluxo de Gravação.	298
Figura 5.5 - Início da Instalação do Programa de Gravação de Imagem.	28
Figura 5.6 - Captura de Imagem.	299

LISTA DE TABELAS

Tabela 5.1 – Custo do Projeto	23
-------------------------------------	----

LISTA DE ABREVIações E SIGLAS

CFTV	Circuito Fechado de TV;
TEK1	Sistema de Gravação de Visualização de Imagem;
USB	Universal Serial Bus;
IVP	Infravermelho Passivo;
IVA	Infravermelho Ativo
TX	Transmissão de sinal;
RX	Recepção de sinal;
MO	Microondas;
RCA	Radio Corporation of América;
S-VHS	Super Video Home System;
RF	Rádio Frequência;
BIT	Binary digiT;
UART	universal asynchronous receiver transmitter;
DB-9	Conector Serial, padrão RS-232, com nove pinos;
RS-232	Padrão da Interface Serial;
AVI	Audio Video Interleave;
CMOS	complementary metal-oxide-semiconductor;
VB	Visual Basic;
IDE	Integrated Development Environment;
DAO	Data Access Object;
RDO	Remote Data Object;
ADO	ActiveX Data Objects;

CAPITULO 1 – INTRODUÇÃO

Hoje em dia, a segurança e a integridade de empresas e residências são fatores de grande preocupação. Estudos e investimentos nessa área são cada vez mais constantes, considerando o alto índice de violência que existe nos dias de hoje.

A segurança por meio eletrônico tem sido usada cada vez mais, por obter uma melhor qualidade e precisão na detecção e na captura de intrusos. A idéia de se obter um sistema de monitoramento que permita se ter menos falhas de natureza humana, é o foco do momento, se mostrando ainda ser um campo bastante promissor.

Graças aos sistemas de monitoramento e mapeamento que são empregados hoje, tornou-se mais simples de localizar e abordar os diversos tipos de negligência que vem ocorrendo nas empresas e até mesmo nas residências. O custo elevado ainda continua sendo um grande empecilho para se obter um sistema de segurança. Por isso o projeto tem como um de seus focos, montar um módulo de detecção que tenha um baixo custo.

O objetivo desse projeto acadêmico é apresentar um sistema de monitoramento que seja capaz de fazer o monitoramento de um ambiente, com gravação da detecção de presença pelo sistema. Este sistema de monitoramento é acionado via interrupção de feixe de luz infravermelho. Ou seja, via detecção de presença. Uma vez que o sistema seja acionado, automaticamente é iniciado, a gravação de vídeo na área monitorada.

O protótipo de monitoramento por um feixe de luz é composto pelas seguintes ferramentas.

- Linguagem de programação assembly e VB;
- Microcontrolador;
- Display;
- Buzzer;
- Conversor serial usb;
- Câmera;
- Laser Infravermelho;

1.1 Estrutura do Trabalho

O trabalho está organizado em seis capítulos. Os primeiros quatro capítulos fazem a apresentação do tema do projeto, fornecem o embasamento teórico do trabalho e as tecnologias utilizadas. No capítulo seguinte é apresentado o protótipo do sistema de monitoramento e o último capítulo traz as considerações finais e propostas futuras. A organização detalhada é descrita a seguir:

Capítulo 1: faz-se a introdução do trabalho.

Capítulo 2: são apresentados os aspectos básicos do sistema, tais como: CFTV, TEK1 e luz infravermelha.

Capítulo 3: é descrito o referencial tecnológico que é abordado na monografia.

Capítulo 4: nesse capítulo são mostrados os hardwares como: microcontrolador, sensor infravermelho, circuito eletrônico utilizados em meu projeto.

Capítulo 5: nesse capítulo são mostrados os softwares que foram desenvolvidos em assembly visual basic.

Capítulo 6: são apresentadas as conclusões, resultados, dificuldades encontradas no projeto e as sugestões de trabalhos futuros.

CAPITULO 2 – FUNDAMENTAÇÕES TEÓRICAS

Nesse capítulo serão vistos modelos de tecnologias utilizadas no sistema de monitoramento nas empresas e residências, qual a vantagem e desvantagem dos sistemas mencionados. Diferenças existentes no espectro eletromagnético, mencionando as categorias de luz infravermelho.

2.1 Sistemas de Monitoramento

A necessidade em resguardar e manter a segurança privada faz com que as pessoas busquem métodos de se precaver contra possíveis danos ao patrimônio pessoal.

Entre os equipamentos mais buscados para a defesa do patrimônio estão os circuitos fechados de TV, com uso de câmeras de captação de imagem.

As mesmas podem servir tanto para alerta de invasão quanto para registro de fluxo de pessoas, furtos, danos ao patrimônio, casos em que a gravação pode servir de prova criminal. Normalmente, a maioria das empresas ou mesmo proprietários de residências utiliza equipamentos de CFTV (Circuito Fechado de TV) para segurança e monitoramento de ambientes.

Basicamente, o sistema CFTV usa câmeras conectadas a uma tela de monitor que geram imagens e que são gravadas em um *time lapse* (aparelho tipo videocassete). Pode-se usar um seqüenciador de imagens, caso haja mais de uma câmera captando as imagens e um sensor de movimento (dispositivo de presença) que liga o sistema de gravação ao menor movimento no ambiente.

Na Figura 2.1 apresenta-se um modelo que será proposto para a criação do protótipo.

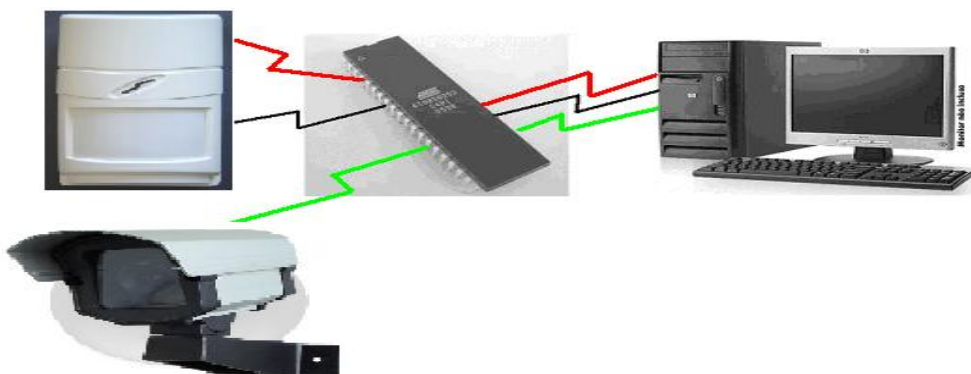


Figura 2.1 - Modelo exemplo de sistema de monitoramento.

Há a possibilidade também de integrar o CFTV ao sistema de vídeo residencial, possibilitando aos moradores da casa ou condomínio ter acesso às imagens do circuito interno em qualquer TV, em um canal paralelo aos demais. Contudo, para isto há a necessidade de se utilizar moduladores de sinal.

Este sistema de monitoramento tem atendido ao usuário quanto à eficiência para a prevenção e controle da segurança. Todavia, a implementação desta tecnologia torna-se onerosa devido ao alto custo do kit, bem como manutenção do sistema.

É válido ressaltar que para implementar este tipo de sistema em um ambiente, há a necessidade de contratar serviços de profissionais ou empresas especializadas em sistemas de segurança.

Com o avanço da tecnologia, vários produtos têm surgido para atender e melhorar os sistemas de segurança. Atualmente, sistemas de gravação e visualização de imagens em computador têm tido uma grande procura pelo baixo custo e as várias vantagens de uso em relação ao sistema CFTV.

No Brasil há o sistema de gravação e visualização de imagens TEK1, que permite gravarem 24 horas por dia sem a utilização de equipamentos especiais. Para isto basta somente instalar o software em um computador com as configurações mínimas e ter uma webcam.

O sistema TEK1, além de ter um custo muito mais acessível, possui vários outros diferenciais em relação ao CFTV:

- Fácil instalação e operação sem a necessidade de técnico especializado;
- Armazenamento e descarte automático de imagens;
- Visualização como filme: em velocidade normal, avançada ou lenta, passo a passo;
- Seleção de imagens com zoom, envio via e-mail ou impressão;
- Processamento compartilhado com outros softwares como Word, Excel, etc.

Outro diferencial do TEK1 em relação a outras tecnologias é a questão da privacidade e da segurança. O usuário faz tudo sozinho, não precisa envolver ninguém. Tem senha, impedindo que não autorizados desativem o TEK1 ou vejam as imagens.

Enquanto que em gravações por CFTV há a necessidade de encaminhar o vídeo a um profissional para se obter imagens específicas de determinado momento de gravação, no sistema de captura em PC, basta apenas escolher a data e delimitar em até décimos de segundos o frame solicitado.

Vale ressaltar que para um aproveitamento melhor desta tecnologia, uma Webcam

de captura de imagem profissional melhoraria a qualidade, bem como a performance em programas de tratamento de imagens. A Webcam que acompanha o kit é de 350Kp (640 x 480) e apresenta qualidade muito boa entre as comercializadas hoje e atende a maioria das necessidades (Tech-faq, 2007).

2.2 Luz Infravermelho

Para entender a visão noturna, é importante compreender um pouco sobre a luz. A quantidade de energia de uma onda luminosa está relacionada ao seu comprimento de onda (comprimentos de onda mais curtos possuem maior energia). Na luz visível, o violeta possui mais energia e o vermelho possui menos. Próximo do espectro da luz visível se encontra o espectro infravermelho.

A Figura 2.2 ilustra onde começa a parte visível ao ser humano e à parte onde não é possível que pessoas enxerguem.

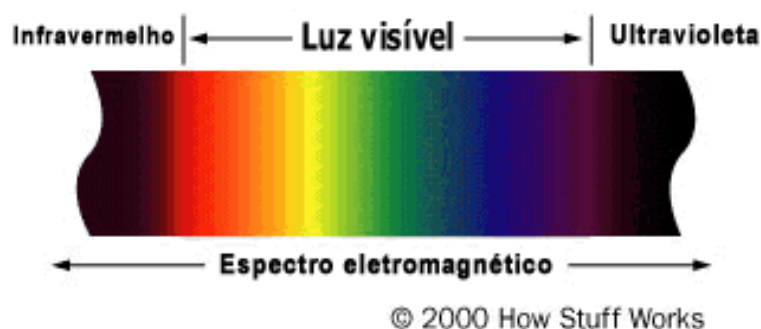


Figura 2.2 - A luz infravermelha constitui uma pequena parte do espectro luminoso.

A luz infravermelha pode ser dividida em três categorias:

- **Infravermelho próximo** (IV próximo): mais próximo da luz visível, o IV próximo possui comprimentos de onda que alcançam de 0,7 a 1,3 **mícrons** ou de 700 a 1.300 bilionésimos de metro;
- **Infravermelho médio** (IV médio): o IV médio possui comprimentos de onda que vão de 1,3 a 3 mícrons. Tanto o IV próximo quanto o IV médio são usados por uma variedade de dispositivos eletrônicos, incluindo os controles remotos;
- **Infravermelho térmico** (IV térmico): ocupando a maior parte do espectro infravermelho, o IV térmico possui comprimentos de onda na faixa de 3 até mais de 30 mícrons.

A diferença fundamental entre o IV térmico e os outros dois é que o IV térmico é emitido por um objeto em vez de ser refletido por ele. A luz infravermelha é emitida por um objeto devido ao que acontece no nível atômico (UOL, 2008).

CAPITULO 3 – REFERÊNCIAL TECNOLÓGICO

Nesse capítulo, será abordado o conceito de sensores suas características e alguns dispositivos que são utilizados atualmente. Modelos de webcam, tipos de conectores existentes e placa de captura também serão vistos.

3.1 Sensores

São dispositivos usados para detectar, medir ou gravar fenômenos físicos tais como calor, radiação etc, e que responde transmitindo informação, iniciando mudanças ou operando controles. Considerem agora outra definição de sensores: são dispositivos que mudam seu comportamento sob a ação de uma grandeza física, podendo fornecer diretamente ou indiretamente um sinal que indica esta grandeza. Quando operam diretamente, convertendo uma forma de energia em outra, são chamados transdutores. Os de operação indireta alteram suas propriedades, como a resistência, a capacitância ou a indutância, sob ação de uma grandeza, de forma mais ou menos proporcional. O sinal de um sensor pode ser usado para detectar e corrigir desvios em sistemas de controle, e nos instrumentos de medição, que freqüentemente estão associados aos sensores de condutividade de malha aberta (não automáticos), orientando o usuário (FACENS, 2006).

Características dos sensores

- Linearidade: é o grau de proporcionalidade entre o sinal gerado e a grandeza física. Quanto maior, mais fiel é a resposta do sensor ao estímulo. Os sensores mais usados são os mais lineares, conferindo mais precisão ao SC. Os sensores não lineares são usados em faixas limitadas, em que os desvios são aceitáveis, ou com adaptadores especiais, que corrigem o sinal.
- Faixa de atuação: é o intervalo de valores da grandeza em que pode ser usado o sensor, sem destruição ou imprecisão (FACENS, 2006).

3.1.1 - Sensor Magnético ou reed switch

É um sensor utilizado para detectar abertura de portas e janelas. É composto por duas partes, uma pequena caixa plástica que possui no seu interior um êmbolo de vidro onde existem duas lâminas metálicas, milimetricamente afastadas que quando sofrem ação de um campo magnético se fecham, permitindo a circulação de corrente.

O campo magnético é obtido através de um ímã de tamanho próximo do sensor (8x8x40 mm) também encapsulado em uma caixa plástica com abas para sua fixação. A caixa com o reed switch é colocada em um ponto fixo da porta ou janela e tem seus terminais ligados com fios à central de alarme, enquanto o ímã é fixado na parte móvel da porta ou janela. Quando a porta está fechada o ímã fica com o contato fechado. Quando a porta é aberta o contato se abre e informa a central que dispara o alarme. Existem vários formatos de ímãs e encapsulantes para sensores magnéticos, sendo os mais comuns os de Sobrepor conforme explicado acima, o de Embutir, que tem as partes encapsuladas em dois cilindros redondos e o para Porta de Aço, que é composto de um ímã maior e permite que a porta possa balançar ou ter jogo sem que o sensor seja acionado (FACENS, 2006).

3.1.2 - Reed switch

É composto por uma caixa plástica de aproximadamente (1x1x8 cm) onde existe uma lâmina de aço fino com um peso e fica levemente encostada a um contato elétrico. Quando o sensor sofre vibração, os contatos se afastam momentaneamente, acionando o alarme. Nestes sensores existe um parafuso que permite ajustar o nível de vibração que fará acionar o sensor. Seu custo é baixo, mas o mesmo tem uso limitado devido a disparos falsos por variação de temperatura e dilatação do metal e acionamento por vibração indesejáveis e locais com solo instável.

3.1.3 - Sensor Infravermelho Passivo

É composto de um detector de luz infravermelha, uma lente e um circuito eletrônico. É chamado passivo porque não emite, mas apenas detecta movimentação de luz infravermelha na sua área de atuação. A base de seu funcionamento é o detector infravermelho ou PIR, que detecta a variação de luz infravermelha e a transforma numa variação de tensão, interpretada pelo circuito eletrônico.

A Figura 3.1 ilustra um circuito que é responsável pela detecção de luz infravermelho.



Figura 3.1 - Sensor Infravermelho Passivo.

O problema de usá-lo diretamente, sem outros acessórios, é que ele seria ativado quando, por exemplo, houvesse variação de luz solar. Para resolver este problema, foi inventada uma lente chamada "Fresnel", que é uma membrana plástica injetada, que permite a passagem de luz infravermelha e possui várias ondulações ou "mini-lentes" que permitem a detecção da variação da luz infravermelha em pontos pré-determinados.

Quando alguém com corpo quente, que emite luz infravermelha, se movimenta em frente ao sensor o mesmo detecta variações nos pontos pré-fixados fazendo com que o PIR receba vários pulsos da variação de luz infravermelha que interpretados pelo circuito, são detectados como sendo um movimento.

Existem sensores IVP (infravermelho passivo) de vários modelos, com lente para corredor tipo cortina, para pequenas e grandes distâncias. Existe um modelo para uso em locais com excesso de insetos ou pó que é o tipo "dual", que de maneira simplificada, possui dois sensores lado a lado, que dificultam o disparo nestes casos. Este tipo de sensor deve ser usado apenas em ambientes internos, de tamanho máximo de 50 metros quadrados. Deve-se evitar o uso em locais muito quentes e onde haja circulação de ar. Em ambientes muito grandes, ou áreas externas, a circulação de ar quente acaba "enganando" o sensor, causando alarmes indesejáveis (Tech-faq,2007).

3.1.4 - Infravermelho Ativo ou Feixe (IVA)

É assim designado por possuir um circuito que emite luz infravermelha (invisível ao olho humano) e outro que detecta a mesma (RX). Os circuitos devem ser colocados frente a frente, em distância pré definida, ou lado a lado, com o uso de espelhos.

Quando um corpo interrompe o feixe de luz emitido pelo TX, o RX detecta a variação, acionando o alarme. O problema de detectar luz IV, é que existem várias fontes da mesma, tal como o sol, lâmpadas incandescentes e até mesmo o nosso corpo. Para contornar este problema, cada parte deste sensor possui uma lente que concentra o feixe do TX e a direciona à posição do RX, além de fazer com que o TX oscile numa frequência fixa, que possa ser filtrada pelo RX. O problema do IVA normal é que a variação de distância de uso e também influências externas tais como chuva, neblina e o próprio sol fazem variar muito sua sensibilidade, ocasionando disparos falsos, principalmente com seu uso em ambientes externos.

Para melhorar a eficiência existem modelos mais modernos que possuem dois emissores de luz IV de frequência diferente que são interpretados pelo RX, além de filtros óticos especiais para o tipo de luz TX, o que gera resultados bem melhores, porém o seu

custo é bem maior. Estes sensores possuem a vantagem de serem usados em distâncias de até 200m em área externa e a desvantagem de permitir que o intruso passe por cima ou por baixo do feixe sem ser detectado.

3.1.5 - Microondas (MO)

Usa um circuito que irradia microondas de baixa potência e uma antena que detecta a reflexão desta radiação em corpos sólidos. Um circuito eletrônico interpreta esta reflexão e verifica quando existe um corpo sólido se movimentando, ativando o alarme.

O problema do sensor é que a MO pode transpassar corpos sólidos como uma parede ou até detectar movimento de água no subsolo. Para resolver este problema, para alarmes é fabricado um modelo que funciona em conjunto com um IVP normal, que só dispara quando ambos detectam algo ao mesmo tempo. O MO possui um ajuste para a sensibilidade da MO refletida, o que permite ajustar o mesmo para não detectar pequenos animais como cães, gatos e pássaros.

Ele deve ser usado em grandes ambientes como barracões, salões e ambientes externos, desde que não haja árvores ou arbustos na área de monitoramento do mesmo, que faz com que o alarme sejam disparados (Tech-faq,2007).

3.2 Modelos de Webcam

Há vários tipos de Webcam no mercado. Os dois padrões mais usados são aquelas para serem usadas com uma placa de captura e as Webcam USB. Há ainda um tipo para ser conectada na saída serial do microcomputador, que não será tratada aqui por ser de baixa resolução.

A Figura 3.2 apresenta três tipos de webcam onde a do meio possui saída de áudio e vídeo composto e as demais são dispostas por uma saída usb.



Figura 3.2 - Tipos de Webcam.

A Figura 3.3 ilustra um exemplo de saídas diferentes de webcam, onde a primeira vem com alimentação junto com uma saída de áudio e vídeo, a outra possui somente um saída USB.

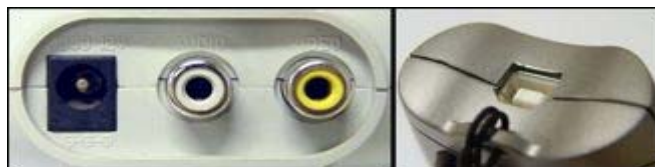


Figura 3.3 - Saídas de áudio e vídeo composto /Conexão USB.

A câmera USB não precisa de uma placa de captura. Ela é ligada diretamente a uma conexão USB do microcomputador. USB é a sigla de Universal Serial Bus, portas disponíveis na maioria dos microcomputadores de hoje, que aceitam a conexão de diversos tipos de dispositivos como mouse, teclado, scanner e câmeras. É de instalação mais simples e também a de menor custo (UNICAMP, 2007).

A Figura 3.4 apresenta um conector macho USB e um conector fêmea USB.



Figura 3.4 - Conector e portas USB.

3.3 Placa de captura de vídeo

Caso o microcomputador tenha uma placa de captura de vídeo, é possível utilizar qualquer câmera com saídas de áudio e vídeo compatíveis com as da placa. As placas e câmeras mais comuns possuem conexões de Áudio/Vídeo Composto (RCA), S-VHS, e RF (rádio frequência). A placa de captura possui a vantagem de receber sinais de vídeo de outros aparelhos além de câmeras, como vídeoK7 e DVD. A desvantagem do uso de

placa de vídeo (mais a câmera) é que é uma opção mais dispendiosa do que a opção de uma câmera USB.

A Figura 3.5 apresenta um placa de vídeo comum, onde temos uma saída BNC, RCA, VHS e composto também com saída de áudio.

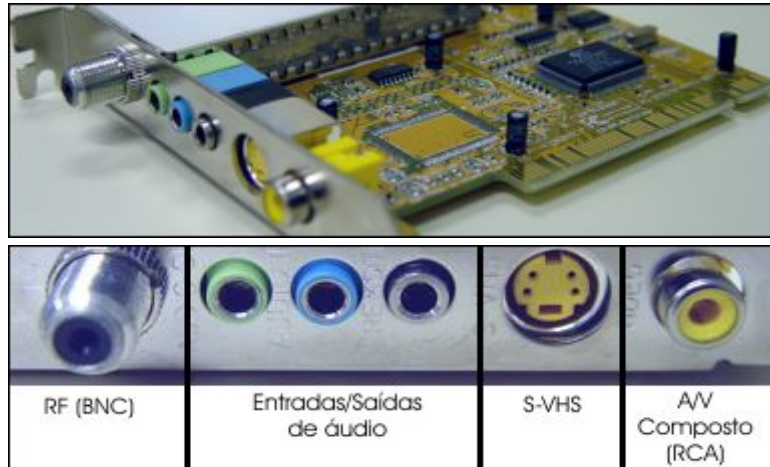


Figura 3.5 - Placa de Captura.

A placa de captura possui um conector para cabos coaxiais BNC. Este conector é utilizado em aplicações de rede de computadores, no transporte de sinais de aparelhos de medição de altas-frequências (osciloscópios por exemplo) e no transporte de sinais de vídeo (imagem) em aplicações profissionais. Possui entradas e saídas de áudio, entradas S-VHS para reprodução de vídeos e uma entrada RCA utilizada para a realização de telecomunicações.

CAPITULO 4 – DESCRIÇÃO DO HARDWARE

Neste capítulo serão abordados todos os componentes envolvidos na elaboração deste projeto. Serão mencionados todos os dispositivos que foram utilizados na criação do módulo de detecção e periféricos como webcam e conversor usb-serial.

O sistema desenvolvido tem como objetivo perceber a incidência ou ausência do feixe de luz infravermelho e gravar o vídeo da pessoa ou objeto que interromper o feixe luminoso.

A Figura 4.1 ilustra o fluxo do modulo de detecção.

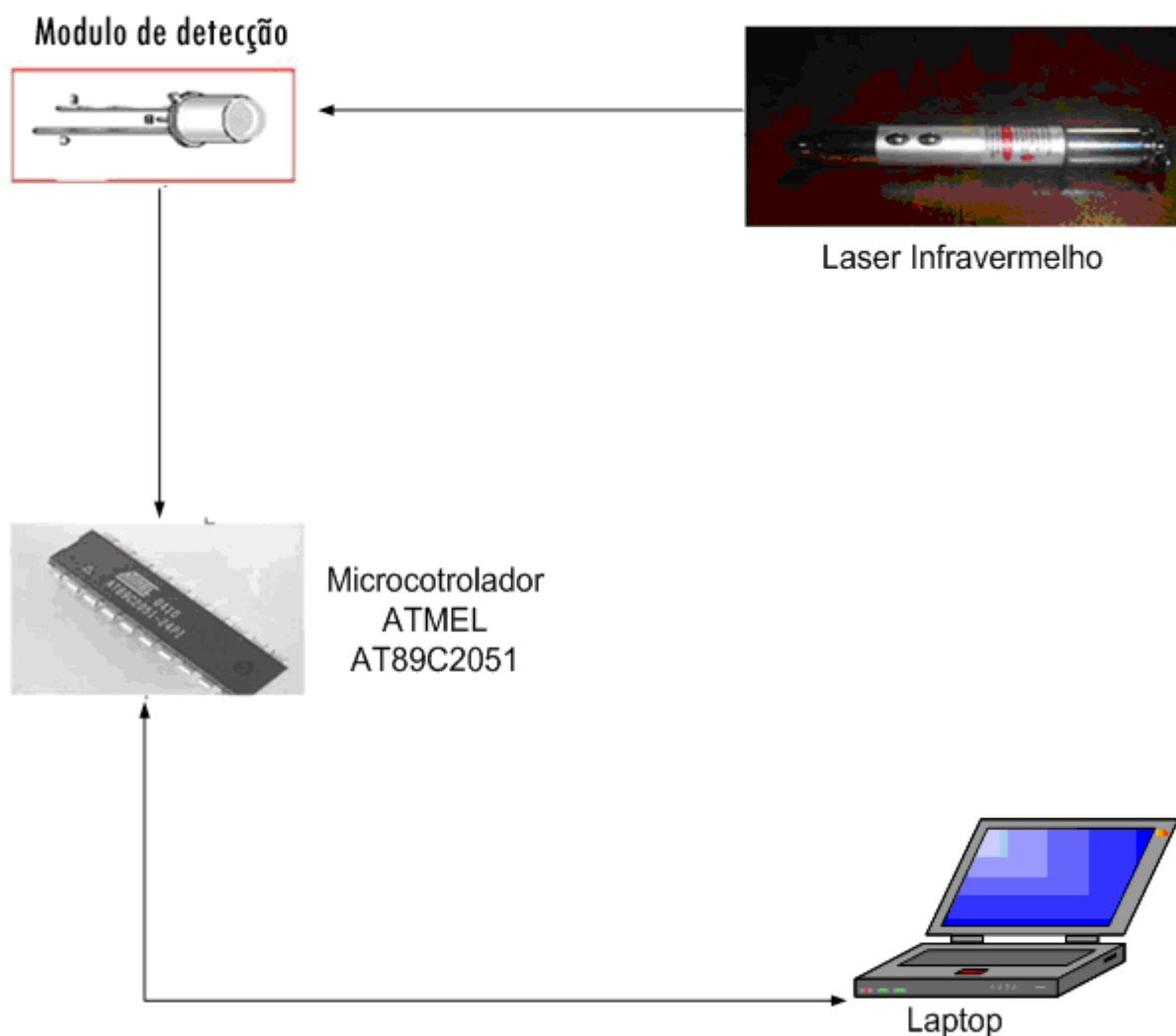


Figura 4.1 - Esquemático do Módulo de Detecção.

Na Figura 4.1, Pode-se ver de que maneira o fluxo segue. Primeiramente ocorre a incidência de um feixe de luz infravermelho no sensor foto transistor, responsável por emitir um sinal ao microcontrolador. Caso haja alguma interrupção, o microcontrolador fica responsável por emitir um sinal ao buzzer que irá soar um alarme, ao mesmo tempo emitirá um vídeo da pessoa ou objeto que foi filmado a partir da porta serial.

A Figura 4.2 ilustra o módulo de detecção e seus dispositivos.

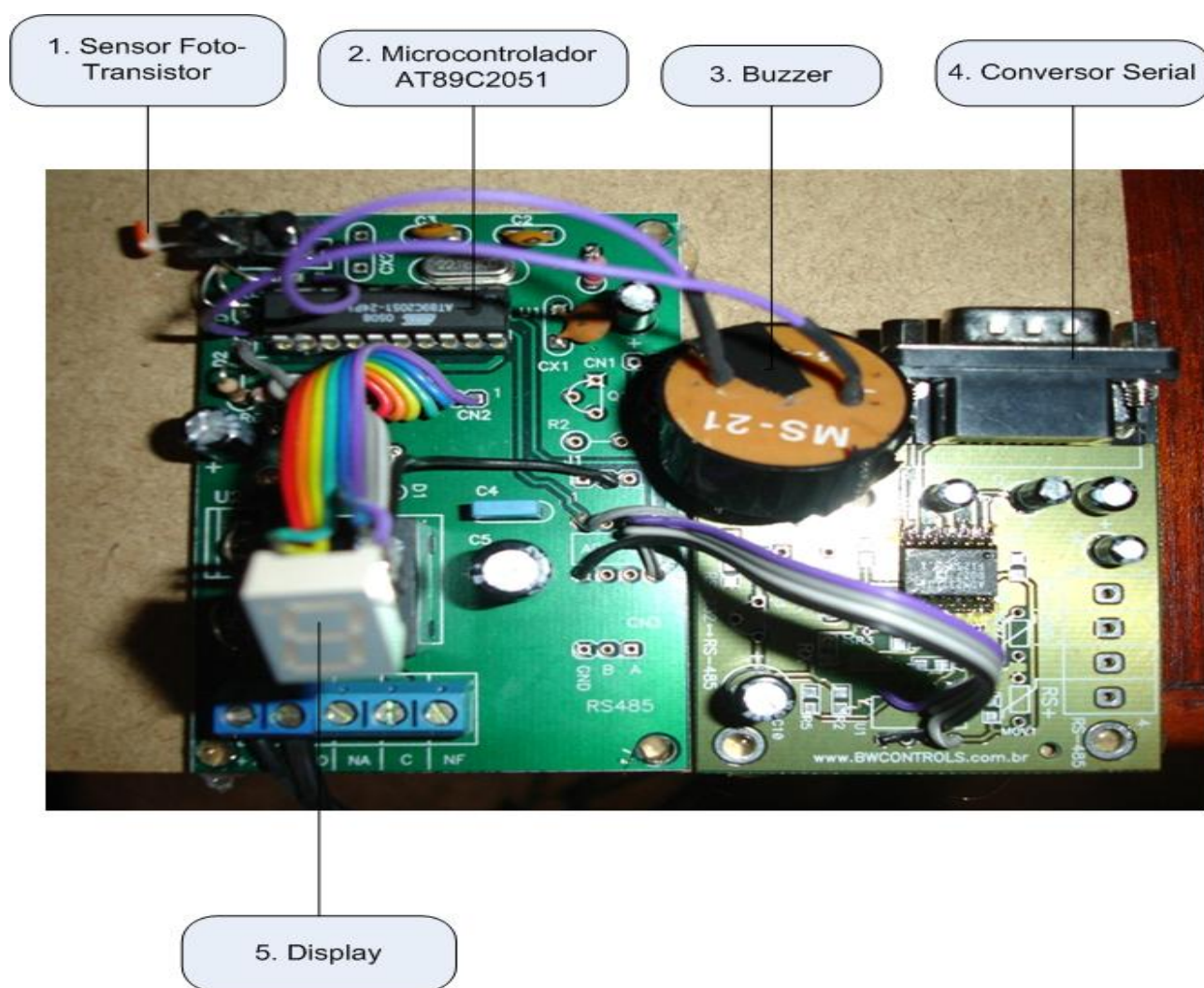


Figura 4.2 - Circuito Desenvolvido.

No desenvolvimento do protótipo de monitoramento foram utilizados os seguintes dispositivos:

1. sensor foto-transistor: função de converter energia luminosa em energia elétrica, emitido assim um sinal ao microcontrolador.

2. Microcontrolador Atmel AT89C2051: já que o mesmo possui a opção de gravar programas, baixo custo e de fácil localização.

3. Buzzer: dispositivo de sinalização normalmente encontrado em automóveis e equipamentos domésticos, cuja função é soar um alarme caso ocorra um interrupção no feixe luminoso.

4. Conversor Serial: manter comunicação entre a CPU e o módulo de detecção.

5. Display: utilizado para aumentar ou diminuir a sensibilidade em que o feixe luminoso pode ser interrompido 0-9, onde o “0” representa maior sensibilidade e o “9” menor sensibilidade.

4.1 Apresentação do Circuito Eletrônico

O circuito possui um microcontrolador da Atmel AT89C2051, qual fornece um ambiente de desenvolvimento integrado e compatível com a família do microcontrolador 8051.

A Figura 4.3 mostra o projeto do circuito que foi elaborado.

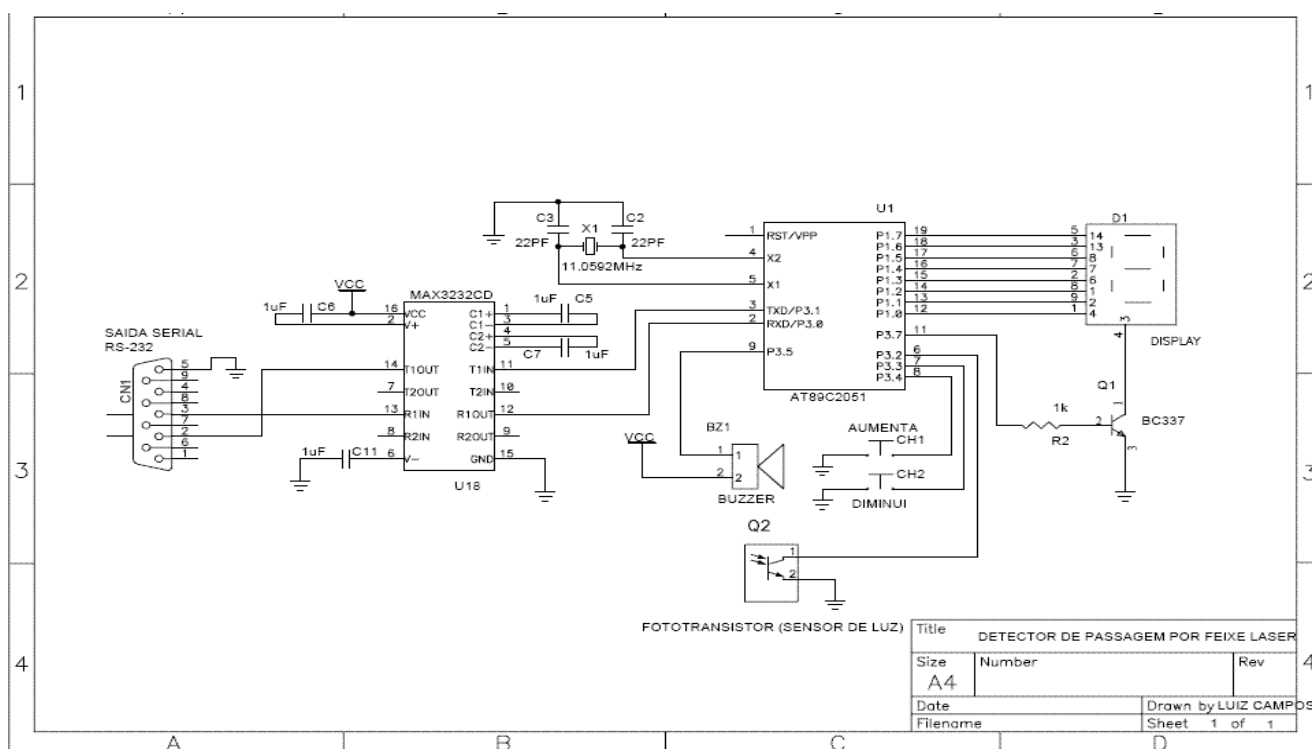


Figura 4.3 - Layout do Circuito.

Portas utilizadas do microcontrolador Atmel AT89C2051:

- 1.0 á 1.7: são usadas para conectar o display, onde a porta 1.0 e 1.1 servem como entrada positiva(AIN0) e entrada negativa(AIN1), enquanto as portas 1.2 á porta 1.7 são utilizadas para entradas externas.
- 3.0 e 3.1 são a RXD (Porta de entrada serial) e TXD (Porta de saída serial) respectivamente. Onde são conectadas ao conversor serial.
- 3.2 INT0 (Interrupção externa 0). Onde é conectado sensor foto transistor.
- 3.3 e 3.4 são INT1 (Interrupção externa 1) e T0 (timer 0 de entrada externa) respectivamente. Onde são conectados os dispositivos para aumentar ou diminuir a sensibilidade.
- 3.5 T0 (timer 0 de entrada externa). Usado para conectar o buzzer.
- 3.7 Utilizada para terra.
- RST Utilizado para redefinir as entradas.
- XTAL1 Inversor e amplificador de entrada para o relógio interno.
- XTAL2 Inversor e amplificador de saída para o relógio interno

4.2 Características do Microcontrolador Atmel AT89C2051

A Figura 4.4 ilustra o microcontrolador utilizado no modulo de detecção.

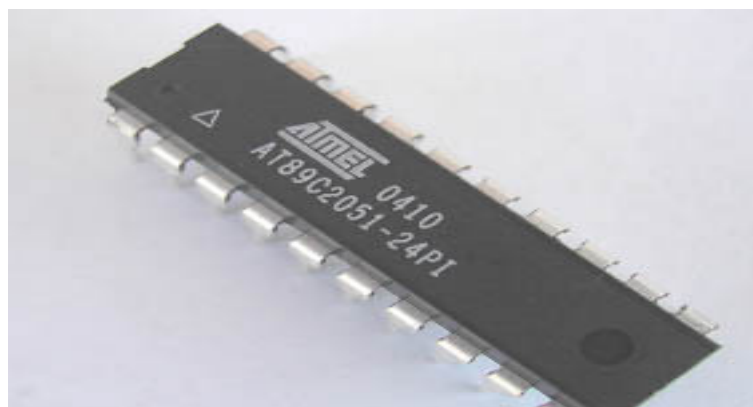


Figura 4.4 - Microcontrolador Atmel AT89C2051.

O AT89C2051 é um microprocessador de 24 pinos compatível com 8051 com 2k de memória flash. O PG2051 apaga programas e verifica o chip AT89C2051 em 6 segundos. Como o AT89C2051 possui memória flash, eles podem ser reprogramadas quantas vezes forem necessárias.

CARACTERÍSTICAS:

- Compatível com MCS ®-51Products ;
- 2K Bytes de memória flash reprogramáveis;
- 2.7V para 6V gama de funcionamento ;
- Dois de nível Program Memory Lock ;
- 6 fontes de interrupção;
- Canal UART programável;
- Baixo custo de energia idle e power-down;

4.3 Interface Serial

A comunicação serial ou interface serial é uma porta de comunicação utilizada para se conectar a outros equipamentos de hardware. Ela se encontra presente nos computadores desde o início da década de 80 e se caracteriza por transmitir e receber um bit de cada vez em seqüência preestabelecida e pré-prolongada (Vasconcelos, 2002). No circuito temos um conversor serial para que o computador possa receber as imagens vindo do microcontrolador.

O padrão utilizado no projeto é o RS-232 uma interface do tipo UART (Universal Asynchronous Receiver and Transmitter) que opera no modo assíncrono, o que indica ao receptor que será enviado um dado, ou seja, a transição de 1 para 0 indica o início da transmissão. Este padrão ainda especifica dois conectores um com 9 pinos (DB-9) e outro com USB (*Universal serial bus*).

A Figura 4.5 ilustra um conversor serial-usb.



Figura 4.5 - Conversor Serial USB.

Foi preciso a utilização desse conversor, devido à ausência de porta serial no notebook a ser utilizado no projeto, sendo mais comum hoje em dia a existência de entradas USB .

4.3.1 - Formato dos Dados Transmitidos

Os dados transmitidos pela porta serial funcionam de acordo com o diagrama mostrado na Figura 4.6. Quando está em repouso ele fornece uma tensão correspondente ao bit 1, ao iniciar a transmissão dos dados é acionado o primeiro bit 0, chamado de start bit. Os dados então começam a ser transmitidos, um bit de cada vez, podem ser transmitidos 5, 6, 7 ou 8 bits dependendo de como a UART esteja programada. Terminados os bits de dados, é enviado um bit opcional de paridade, e por ultimo, um bit finalizador chamado de stop bit, que tem sempre o valor 1. Imediatamente após o stop bit, pode ser enviado o start bit do dado seguinte (Vasconcelos, 2002).

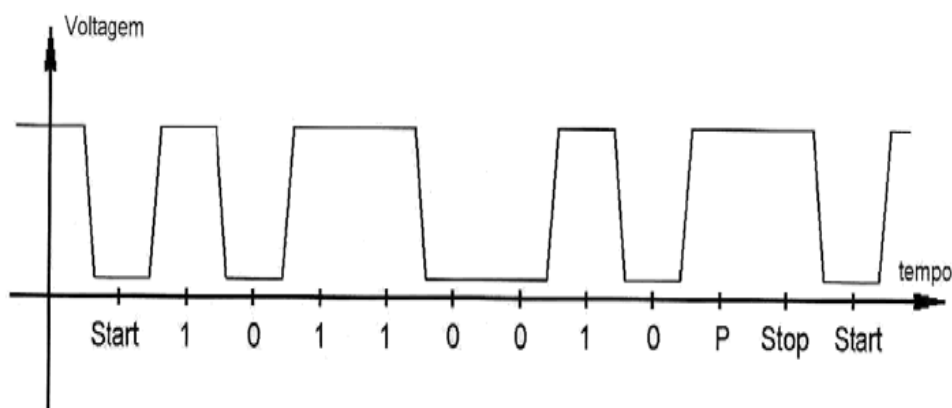


Figura 4.6 - Transmissão de uma Porta Serial.

4.3.2 - Comunicação Serial do Projeto

A transmissão serial do projeto utilizou os pinos de transmissão de dados (TX), pino 3, e recepção de dados (RX) pino 2. O aterramento, caso ocorra nível de tensão acima de 5 volts é feito pelo pino 5 (GND). .

A Figura 4.7 representa o que significa cada entrada serial de um conector DB9.

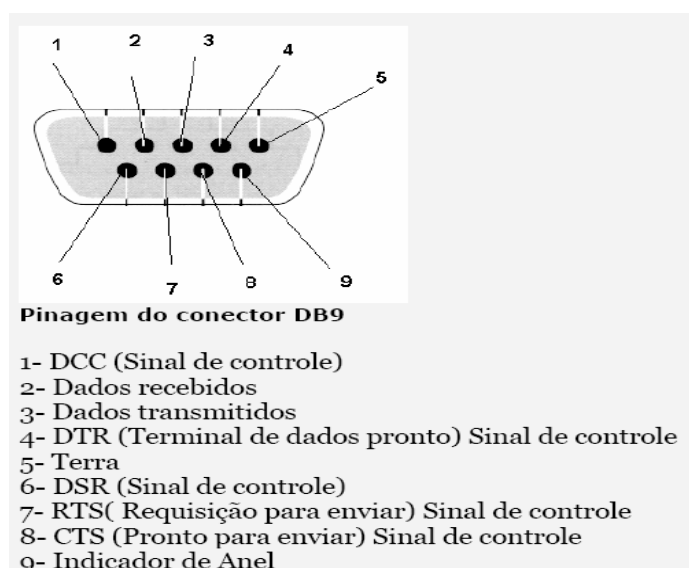


Figura 4.7 - Configuração do Conector Serial.

Para reconhecer a porta serial, durante o desenvolvimento é necessário especificar todos os parâmetros de comunicação com a porta serial como: qual porta ele fará a comunicação (COM1, COM 2).

4.4 - Câmera Utilizada para Capturar a imagem

A câmera que está sendo usada é uma câmera web comum, marca kemex AW-1035. A conexão é feita através da USB, Abaixo segue a figura da câmera a ser usada e algumas características da mesma.

A Figura 4.8 mostra uma webcam comum na qual será usado no projeto.



Figura 4.8 - Câmera kemex AW-1035.

Algumas das características obtidas no manual da webcam seguem abaixo:

- ✓ Sensor de imagem: CMOS colorido 350 pixels ;
- ✓ Formato de imagem: 640 x 480 pixels;
- ✓ Formato de captura de imagem: JPG e BMP;
- ✓ Função de compressão AVI para vídeos;
- ✓ Velocidade máxima de captura de imagem AVI: 15 fps VGA;
- ✓ Interface USB versão 1.1;
- ✓ Não requer alimentação externa;
- ✓ Alimentação através da porta USB – 5 volts/5-mA;

4.5 - Dispositivo Infravermelho

O feixe de luz infravermelho é projetado direto no sensor fototransistor, que fará a detecção de pessoas ou objetos que interrompam o feixe luminoso, acionando um alarme através do buzzer e emitindo a imagem direta ao computador. Segue abaixo o dispositivo responsável por emitir a luz infravermelho.

A Figura 4.9 mostra o dispositivo de luz infravermelho a ser usado no projeto.



Figura 4.9 - Dispositivo que Emite Luz Infravermelho.

Foi utilizado um dispositivo doméstico para a emissão do feixe de luz infravermelho, por não ser de fácil localização algo que seja mais profissional.

4.6 - Protótipo Desenvolvido

O desenvolvimento do protótipo integra os seguintes componentes e tecnologias: laser infravermelho, sensor fototransistor, microcontrolador da Atmel AT89C2051, buzzer responsável por soar o alarme e programação na linguagem assembly. Entre os componentes citados, o microcontrolador tem o papel principal, pois é nele que serão armazenadas todas as regras do código desenvolvido em assembly, como:

- ♦ Comunicação entre o circuito e o computador;
- ♦ Detecção da interrupção do feixe de luz infravermelho;
- ♦ Buzzer, responsável por fazer soar o alarme quando hover alguma interrupção no feixe.

NA Figura 4.10 é apresentada a fotografia do protótipo desenvolvido. Pode-se observar que o circuito está conectado ao computador. E essa comunicação é feita pela porta serial.



Figura 4.10 - Protótipo desenvolvido.

A linguagem de programação utilizada no desenvolvimento do microcontrolador foi desenvolvido em assembly, responsável por detectar a interrupção do feixe infravermelho, e fazer o alarme soar.

Foi utilizado a linguagem programação visual basic para fazer a gravação dos vídeos quando houver alguma interrupção no feixe de luz infravermelho.

O código-fonte do assembly e visual basic, são apresentados completos nos Apêndices 1 e 2.

CAPITULO 5 – DESCRIÇÃO DO SOFTWARE

O projeto desenvolvido está dividido em duas etapas: a primeira etapa apresenta o circuito eletrônico, onde são apresentados os componentes responsáveis por tratar o sinal infravermelho e converte-lo através de uma porta serial ao computador. Na segunda etapa é apresentado o sistema de gravação, responsável por gravar e armazenar as imagens a serem gravadas. Junto com a apresentação de ambas as etapas são descritas as tecnologias e ferramentas utilizadas para seu desenvolvimento.

Os componentes que fazem parte do projeto foram escolhidos visando à qualidade, simplicidade, rapidez e baixo custo. São ferramentas consolidadas do mercado e que possuem vasta documentação. Na Tabela 1, tem-se a relação de todas as ferramentas e tecnologias utilizadas no decorrer do projeto e o custo de cada. Ressaltando que esses preços servem como base para quem venha a utilizar algum desses componentes ou ferramentas.

Tabela 1 – Custo do Projeto

Componentes e Ferramentas	Custo	Observação
Sistema operacional dos	R\$ 0,0	Utilizado para desenvolvimento do microcontrolador
Visio 2003	R\$ 101,08	Utilizado na criação do fluxograma do projeto e modelagem do sistema. <i>Fonte do preço: www.submarino.com.br</i>
Conversor serial - USB	R\$ 45,0	Comunicação do circuito com o computador
Web Cam	R\$ 35,0	Responsável por gravar as imagens.
Laser Infravermelho	R\$ 20,00	<i>Fonte do preço: www.mercadolivre.com.br</i>
Circuito Eletrônico	R\$150,00	Todo circuito eletrônico utilizado no projeto
Custo Total	R\$ 351,08	

5.1 – Fluxo de Comunicação

O fluxo de comunicação do módulo de detecção começa a partir do momento em que há algum tipo de interrupção no feixe de luz infravermelho. O feixe de luz é incidido no sensor foto transistor, caso não ocorra nenhuma intervenção nada ocorre, tendo a não incidência do laser, o sensor foto transistor envia um sinal elétrico ao microcontrolador onde fica responsável por tratar o sinal e enviar um outro sinal para o conversor serial.

O módulo de detecção também possui uma função de aumentar ou diminuir a sensibilidade, que varia de 0-9. Quando estiver no 0 o tempo de resposta é muito mais rápido, com o aumento da sensibilidade, aumenta-se o tempo de resposta para que o módulo de detecção comece a gravar. O código se encontra no apêndice 1.

A Figura 5.1 ilustra o fluxo de comunicação

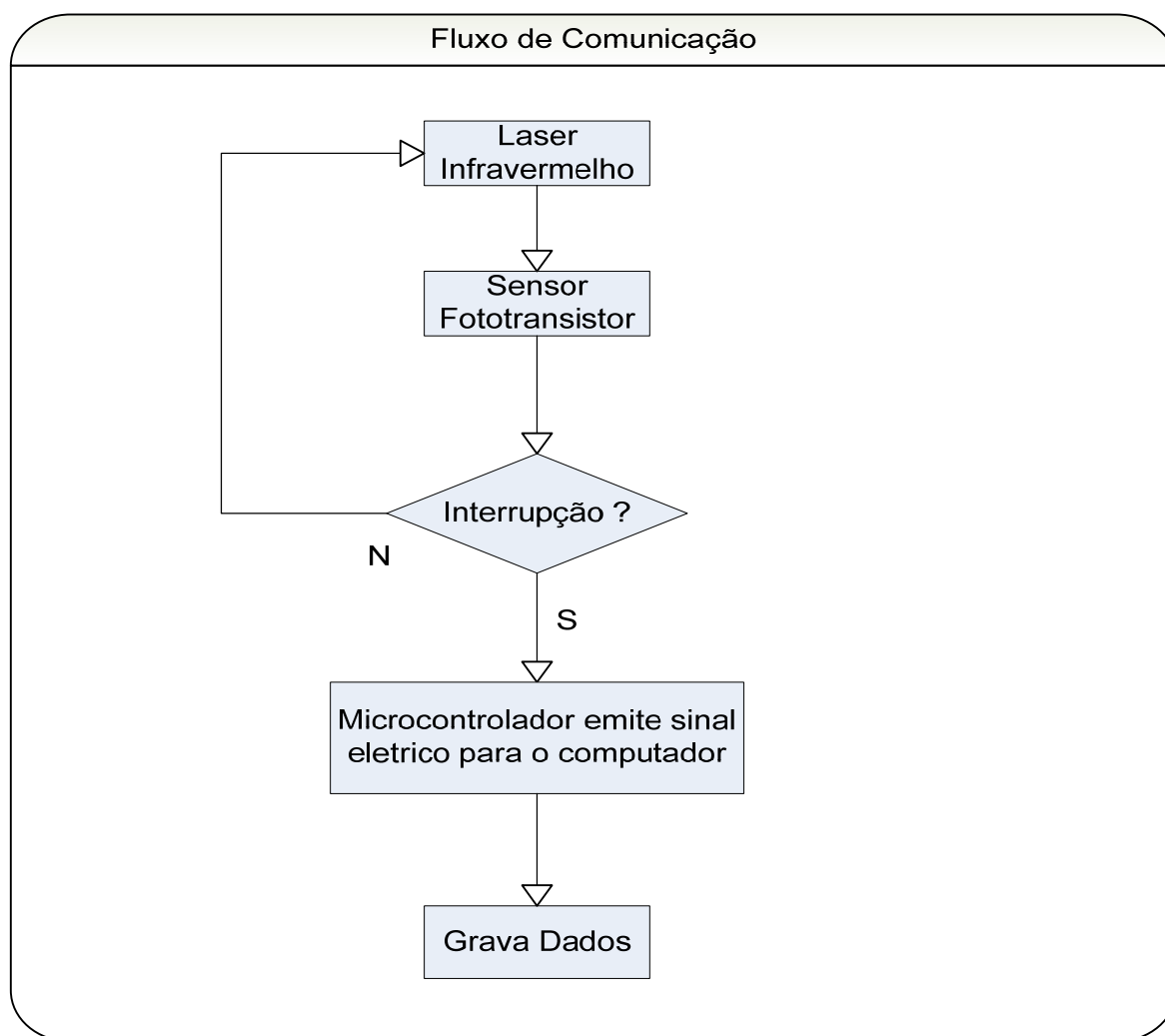


Figura 5.1 - Fluxo da Comunicação

O vídeo é gravado no diretório c:\ em um arquivo chamado capture.avi. As gravações sempre são realizadas no mesmo arquivo, ou seja, quando houver um novo vídeo, irá sobrescrever o mesmo gravado anteriormente.

5.1.2 - Fluxo de Comunicação Serial

O microcontrolador envia um sinal ao conversor serial, o mesmo comunica-se com o laptop a partir da porta COM1, ao receber o sinal o software de gravação dispara a gravação de vídeo.

A Figura 5.2 ilustra o fluxo de comunicação serial.

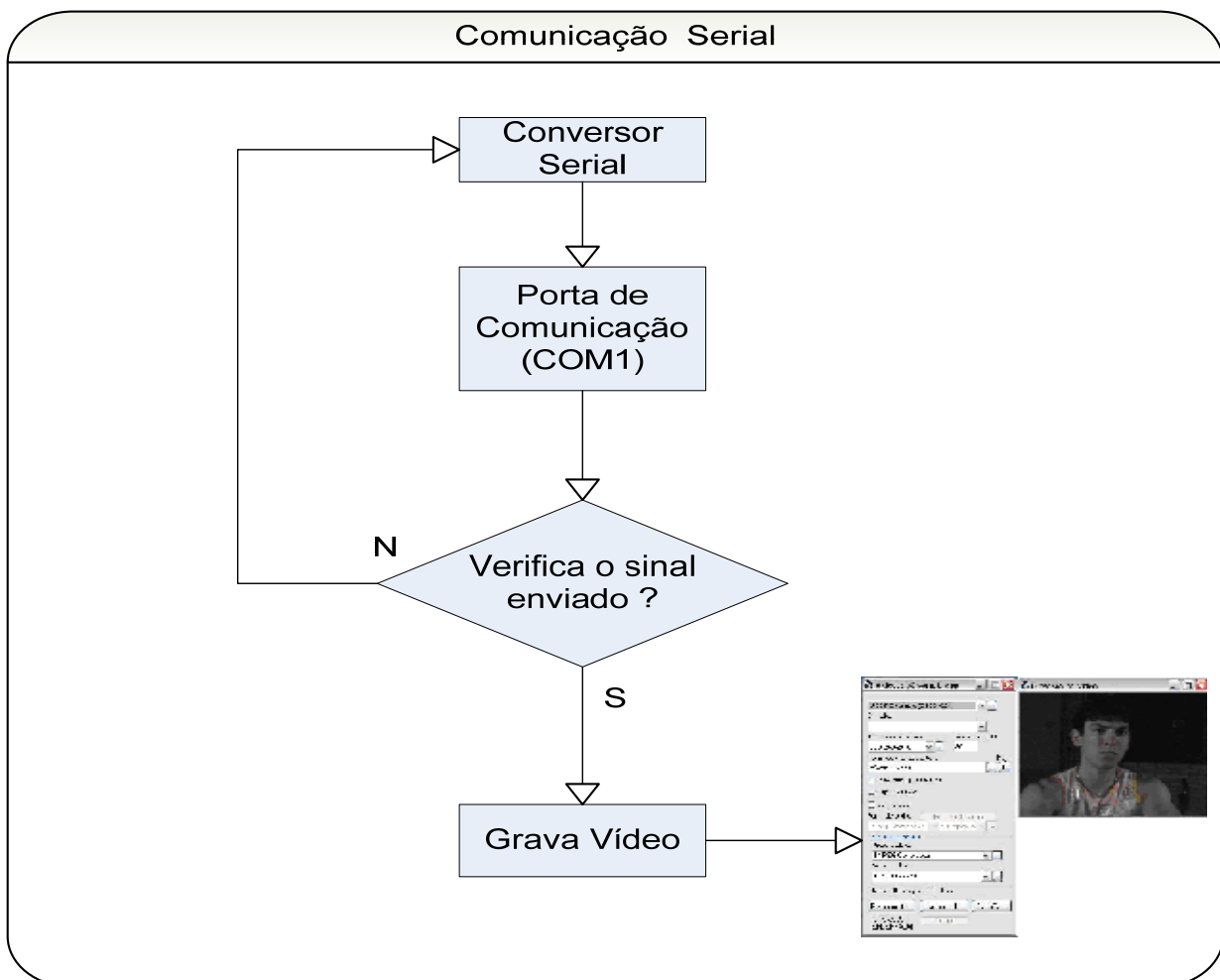


Figura 5.2 - Fluxo de Comunicação Serial.

5.2 - VideoCapX

O VideoCapX é um aplicativo que reproduz vídeo podendo ser feitas modificações em seus códigos, é elaborado a partir do controle active X(ocx). Uma ocx representa uma biblioteca de um determinado programa, onde se pode personalizar da maneira que bem entender. No meu projeto essa ocx é de um programa de execução de vídeo como o Windows media player.

VideoCapX permite que os desenvolvedores facilmente adicionem um acesso a placas de captura de vídeo digital e câmeras. Ele fornece uma interface para exibir o vídeo na tela, interação com o usuário, controle do formato do vídeo e gravação *full-motion*. VideoCapX trabalha com qualquer dispositivo compatível com o Windows e oferece um controle completo sobre a captura, assim como opções de compressão e de gravação.

Mesmo que você esteja desenvolvendo algo para webcam, um aplicativo de segurança para a rede, um banco de dados com identidades e fotografias, um sistema de monitoramento industrial ou um dispositivo médico, VideoCapX resolve o problema de capturar e salvar imagens em seqüências de vídeo.

Características :

- Suporte a dispositivos compatíveis com WDM/DirectX: câmeras FireWire (DCAM), placas de captura, PCI, Câmeras USB/USB2, dispositivos DV, sintonizadores de TV.
- Captura de quadros simples e seqüências de imagens
- Suporte a codecs de áudio e vídeo
- Sobreposição de gráficos e textos
- Captura de filmes AVI ou WMV
- Transmissão de streams de vídeo Windows Media sobre a rede ou para um servidor WM
- Funcionalidades de videoconferência

A Figura 5.3 representa o início da instalação do VideoCapX.

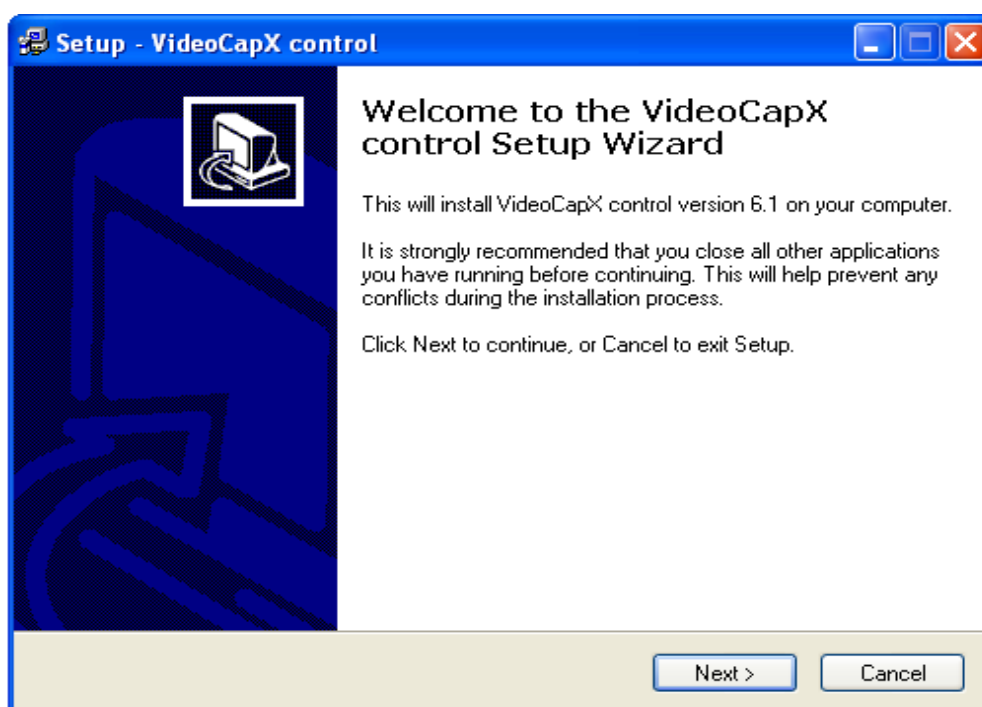


Figura 5.3 - Tela Inicial de Instalação do VideoCapX.

5.3 - Captura de Imagem

Para que sejam gravadas as imagens quando houver interrupção, foi preciso fazer uma adaptação no mesmo código usado no VideoCapX. Essa adaptação foi desenvolvida na mesma linguagem de programação da ocx onde é desenvolvida em visual basic. O código em Visual Basic que se encontra no apêndice 2 somente foi alterado para gravação dos vídeos, que se encontra nas páginas 56 e 57.

Quando o módulo de detecção detecta que houve alguma interrupção, envia um sinal elétrico onde o software de gravação começa a gravar, quando não houver mais interrupção o processo é pausado.

A Figura 5.4 representa o fluxo de gravação.

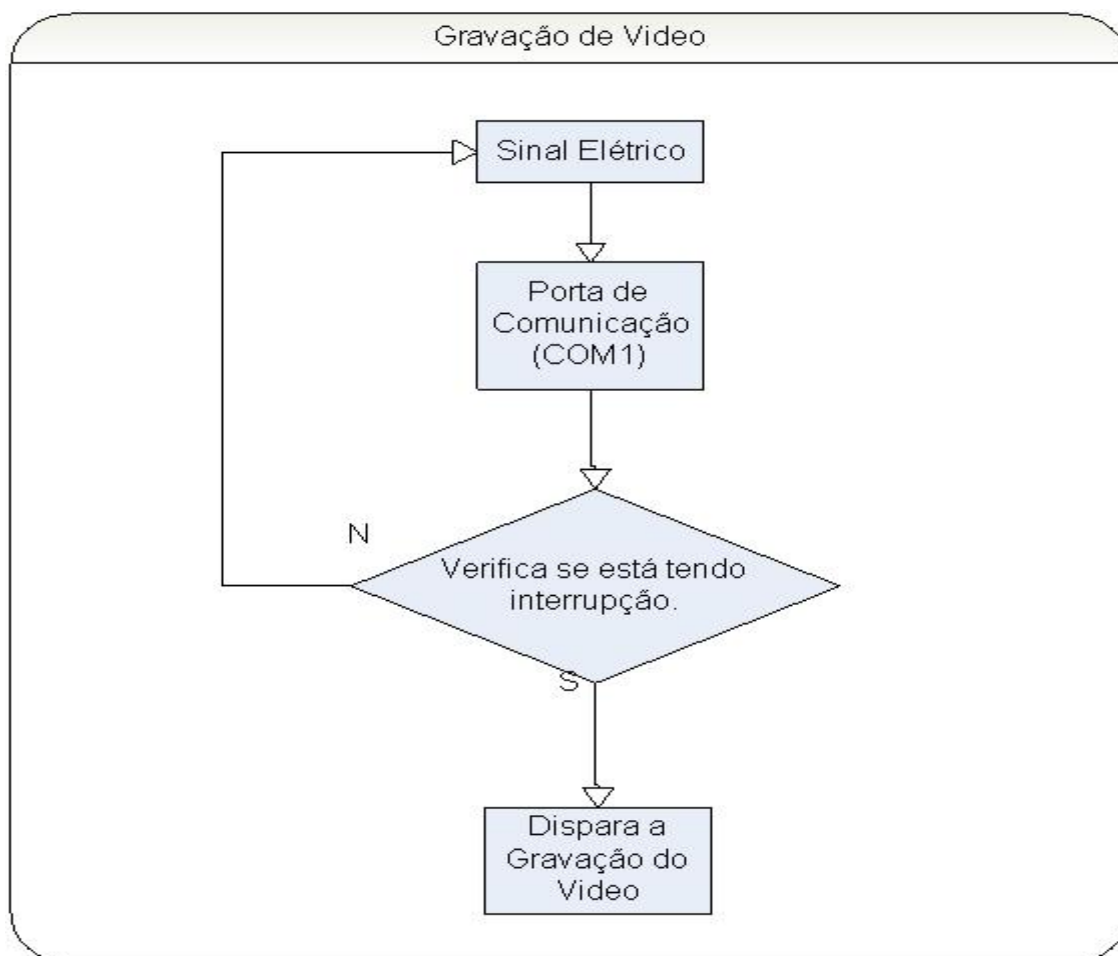


Figura 5.4 Fluxo de Gravação

A seguir será mostrado o processo de instalação do software de captura de imagem. A Figura 5.5 mostra a tela inicial de instalação do programa responsável pela captura da imagem.



Figura 5.5 - Início da Instalação do Programa de Gravação de Imagem.

A Figura 5.6 ilustra a tela de captura de vídeo do aplicativo Vídeo CapX. Por default os vídeos são armazenado no arquivo c:\capture.avi.

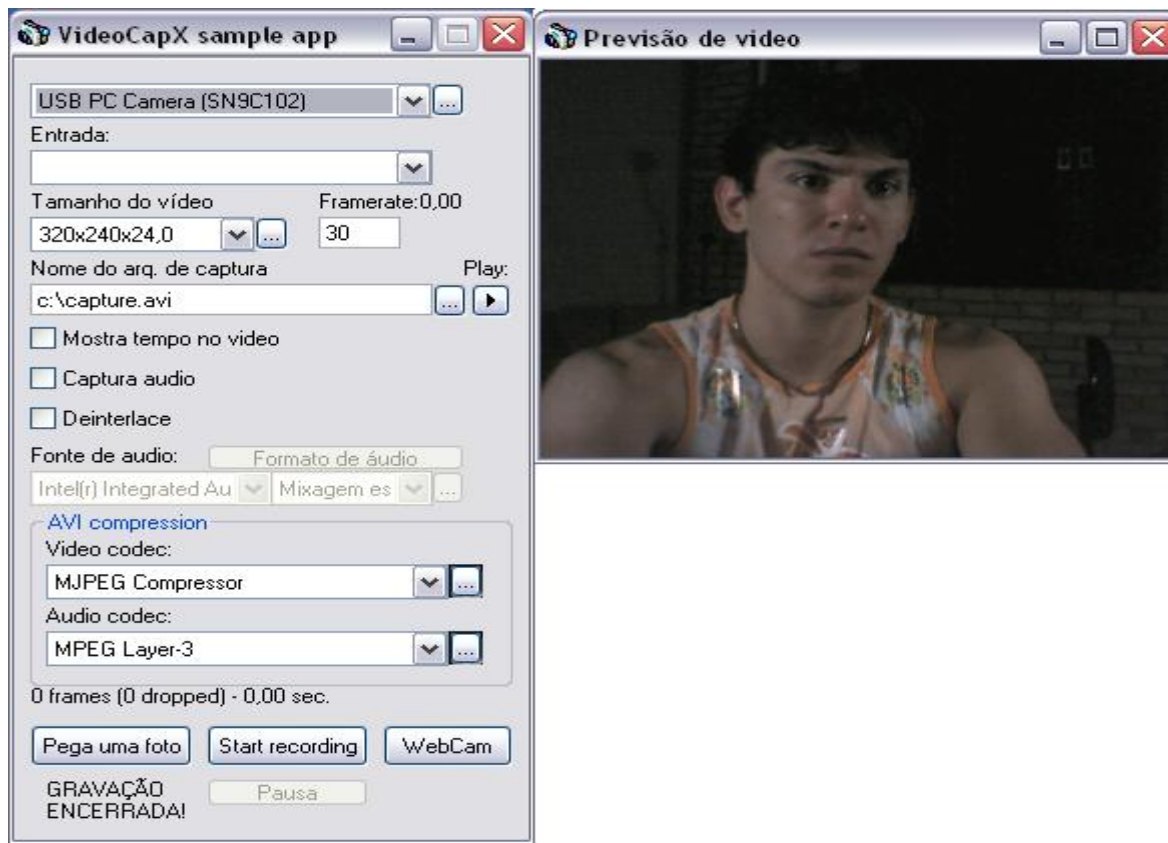


Figura 5.6 - Captura de Imagem.

No Vídeo CapX , temos a opção de escolher o tamanho do vídeo para melhor visualização, alterar o tipo de codec a ser utilizado na gravação, mostrar o tempo de vídeo que foi gravado e por fim capturar o áudio.

CAPÍTULO 6 - CONCLUSÃO

Foi implementado neste trabalho um protótipo de sistema de monitoramento infravermelho por um feixe de luz, conforme objetivo descrito no início do mesmo.

O método utilizado no processo de monitoramento por um feixe de luz, mostrou-se uma alternativa eficiente e apresentou um baixo custo. Uma grande gama de aplicações pode ser implementada com a técnica apresentada. Algumas delas serão citadas na subseção 6.1.

As maiores dificuldades obtidas durante o desenvolvimento do projeto foram: fazer a comunicação serial entre o hardware e o software. Na comunicação serial foi preciso especificar a porta corretamente no sistema de gravação para que houvesse uma comunicação correta, ou seja, não se podia utilizar qualquer porta. Isto demandou tempo para que fosse descoberto.

Apesar de todos os obstáculos, espera-se que esse projeto possa ser referência de ajuda para os próximos formando de Engenharia da Computação que tiverem interesse em fazer algum projeto na área de monitoramento.

6.1 - Propostas Futuras

Várias opções podem ser seguidas a partir do trabalho aqui apresentado. As sugestões serão citadas logo abaixo.

- ✓ Pode-se adicionar mais segura para armazenamento dos vídeos ou imagens capturados pela câmera, utilizando banco de dados robustos como ORACLE, SQL Server, Mysql entre outros.
- ✓ A comunicação entre o circuito e o computador pode ser feita via rede sem fio (wireless) usando criptografia para maior segurança.
- ✓ Desenvolvimento de um sistema que permita que o feixe de luz fique em movimento, assim dificultando algum tipo de mapeamento dos feixes.
- ✓ Fazer um sistema de gravação de vídeo onde após ocorrer a gravação seja enviado um e-mail ou mensagem ao responsável pela segurança do ambiente

REFERÊNCIAS BIBLIOGRÁFICAS

ELS, Rudi van. **Sistema de desenvolvimento para microcontroladores CW552**, CONTROLWARE Automação Comercial, Versão 7, Brasília 2001.

Futurlec Livre [**Home Page**]. 2007. Disponível em:

<http://www.futurlec.com/Atmel/AT89C2051>. Acessado em 13 de agosto de 2007

Rodrigo Benin Ribeiro, Arma Sentinela. Centro Universitário de Brasília – UniCEUB, Faculdade de Ciências exatas e de Tecnologia – FAET, curso de Engenharia da computação 2007.

Linha de Código [**Home Page**]. 2007. Disponível em:

http://www.linhadecodigo.com.br/artigos?id_ac=655. Acessado em: 9 de agosto de 2007.

Mercado Livre [**Home Page**]. 2007. Disponível em:

<http://guia.mercadolivre.com.br>. Acessado em 13 de agosto de 2007.

Nicolosi, Denys E. C. **Microcontrolador 8051 detalhado**. São Paulo: Editora Érica 5ª edição, 2004.

UOL [**Home Page**].2008. Disponível em: <http://eletronicos.hsw.uol.com.br/visao-noturna1.htm>. Acessado em 20 de março de 2008.

UNICAMP [**Home Page**].2007. Disponível em <http://www.unicamp.br>. Acessado em 10 novembro de 2007.

VASCONCELOS, Laércio. **Hardware Total**. São Paulo: Makron Books, 2002

Tech-faq [**Home Page**]. 2007. Disponível em:

<http://www.tech-faq.com/usb.shtml> . Acessado em:9 de agosto de 2007.

```
;=====| E/S |=====
```

```
LDR          BIT    P3.5
BUZZER       BIT    P3.4
RELE        BIT    P3.7
PONTO       BIT    P1.5

SENS        DATA   40H
TIMER_VAR   DATA   41H
TICKS      DATA   42H

FLAG_20SEGUNDOS  BIT    24H.0
FLAG_BUZZER     BIT    24H.1
```

```
;---| ESQUEMA DE LIGACAO NO DISPLAY |-----
;
;           CENT.      DEZ.      UNID.
;
;           A
;           |
;           V
;           +-+
; F-> | | <-B
;           +-+
; E-> | G | <-C
;           +-+ . <- h
;           ^
;           |
;           D
;
; SEG.| PINO
;
; A-> P1.2
; B-> P1.3
; C-> P1.4
; D-> P1.6
; E-> P1.7
; F-> P1.1
; G-> P1.0
; H-> P1.5
;
;
;
;           /-----\
;           -=| INICIO |=-
;           \-----/
;
;
```

```
ORG 0000H                ; Reset
jmp INICIALIZA

;---
ORG 0003H                ; Vetor da interrupcao Externa 0 (INT0)

ajmp INT_EXT0
```

```

;---
ORG 000BH          ; Vetor da interrupcao do TIMER 0
    ajmp INT_TIMER0
;---
ORG 0013H          ; Vetor da interrupcao Externa 1 (INT1)
    ajmp INT_EXT1
;---
ORG 001BH          ; Vetor da interrupcao do TIMER 1
    RETI
;---
ORG 0023H          ; Vetor da interrupcao serial
    RETI
;---

```

INICIALIZA:

;Seta Velocidade serial

```

    clr    TR1          ; Desliga timer 1

    mov    SCON,#01010000B    ; TI setado indica transmissor pronto
                                ; modo 1 ,REN
    mov    TMOD,#00100001B    ; Timer 1 is set to 8 bit auto reload mode
                                ; Timer 0 = contador de 16 bits

    mov    PCON,#0          ; Zera o para baixar para 1/2 do baudrate

    mov    a,#0FAh          ; velocidade = (9600 em 22.1184MHz)

    mov    TH1,a            ; Carrega com o valor de "reload"

    setb   TR1              ; Inicia Timer 1
    clr    ET1
                                ; Velocidade = 9600 bps
    setb   ES                ; Habilita interrupcao serial

    mov    P1,#0ffh
    setb   LDR

    mov    TIMER_VAR,#30
    mov    TICKS,#20

    clr    FLAG_20SEGUNDOS
    clr    FLAG_BUZZER

    setb   EX0                ; Habilita interrupcao externa 0
    setb   IT0                ; Ativa interrupcao no estado 0

    setb   EX1                ; Habilita interrupcao externa 1
    setb   IT1                ; Ativa interrupcao no estado 1

```

```

    clr BUZZER                ; apita o buzzer

    mov a,#2                  ; espera dois segundos
    lcall delay_sec

    setb BUZZER               ; Desliga buzzer

    mov a,#2                  ; espera dois segundos
    lcall delay_sec

    clr BUZZER                ; apita o buzzer

    mov a,#2                  ; espera dois segundos
    lcall delay_sec

    setb BUZZER

    mov SENS,#1                ; Sensibilidade = 1 (Pouco sensivel)
    mov dptr,#disp_table
    mov a,SENS
    movc a,@a+dptr

    mov Pl,a                  ; Imprime no display

; Inicializa timer 0

    clr TR0

    mov TH0,#HIGH(60000)      ; Carrega o byte MSB no timer 1
    mov TL0,#LOW(60000)       ; Carrega o byte LSB no timer 1

    setb TR0                  ; Liga Timer 0
    setb ET0                  ; Habilita interrupcao TIMER 0
    setb EA

START:
    setb BUZZER

S2:
    mov a,#2
    acall delay_ms

    jnb LDR,S2

    mov a,SENS
    mov b,#25
    mul ab

    lcall delay_ms

    jnb LDR,START

ALARME:
    jb FLAG_BUZZER,S2

    setb FLAG_BUZZER
    mov TIMER_VAR,#30
    mov TICKS,#30

    clr TI
    mov sbuf,'#A'
    jnb TI,$

```

```

        sjmp start

;
; Rotina de atendimento a INT0
;

INT_EXT0:

        push ACC
        push PSW
        push dpl
        push dph

DEBOUNCE0:

        mov a,#10
        lcall delay_ms

        jb p3.2,INT0_EXIT

        inc SENS

        mov a,SENS

        cjne a,#11,INT0_EXIT
        mov SENS,#10

INT0_EXIT:

        mov dptr,#disp_table
        mov a,SENS
        movc a,@a+dptr

        mov P1,a                ; Imprime no display

        clr BUZZER

        mov a,#80
        lcall delay_ms

        setb BUZZER

        jnb P3.2,$

        pop dph
        pop dpl
        pop PSW
        pop ACC

        reti

;
; Rotina de atendimento a INT1
;

INT_EXT1:

        push ACC
        push PSW

```

```

push dpl
push dph

```

DEBOUNCE1:

```

mov a,#10
lcall delay_ms

jnb p3.3,INT0_EXIT

clr C
mov a,SENS
subb a,#1
mov SENS,a

jnc INT1_EXIT
mov SENS,#0

```

INT1_EXIT:

```

mov dptr,#disp_table
mov a,SENS
movc a,@a+dptr

mov P1,a                      ; Imprime no display

clr BUZZER

mov a,#80
lcall delay_ms

setb BUZZER

jnb P3.3,$

pop dph
pop dpl
pop PSW
pop ACC

reti

```

```

; |-----|
;--=| Rotina de delay de 1 mS vezes o valor de A |=-
; |-----|

```

DELAY_MS:

```

push ACC
push B
mov B,#0

DD:
djnz B,$          ; ~ 500 uS @ 12 Mhz
djnz B,$          ; ~ 500 uS @ 12 Mhz

djnz ACC,DD

pop B
pop ACC
ret

```

```

; -----|
; --= Delay de 1 segundo vezes o valor de A =-
; -----|

```

DELAY_SEC:

```

    push ACC
    push B

```

```

    mov B,A

```

DDD:

```

    mov a,#250
    lcall DELAY_MS    ; 500mS
    lcall DELAY_MS    ; 1000mS

```

```

    djnz B,DDD

```

```

    pop B
    pop ACC
    ret

```

DISP_TABLE:

```

    DB 00100001B    ; 0
    DB 11100111B    ; 1
    DB 00110010B    ; 2
    DB 10100010B    ; 3
    DB 11100100B    ; 4
    DB 10101000B    ; 5
    DB 00101000B    ; 6
    DB 11100011B    ; 7
    DB 00100000B    ; 8
    DB 10100000B    ; 9
    DB 11111110B    ; -
    DB 11011111B    ; .

```

```

; -----|-----|-----|
; -----= ROTINA DE ATENDIMENTO A INTERRUPCAO -----
; ----- TIMER 0 -----
; -----|-----|-----|

```

INT_TIMER0:

```

    push ACC
    push PSW

```

```

    clr TR0

```

```

    mov TH0,#HIGH(60000)    ; Carrega o byte MSB no timer 1
    mov TL0,#LOW(60000)     ; Carrega o byte LSB no timer 1

```

```

    djnz TIMER_VAR,TIMER0_SAIDA    ; Decrementa variavel , se nao for 0
                                    ; sai direto

```

```

    mov TIMER_VAR,#30

```

```

;-- 1 SEGUNDO

```

```

        cpl PONTO

        jnb FLAG_BUZZER,BUZZER_PULA

        cpl BUZZER

BUZZER_PULA:

        djnz TICKS,TIMER0_SAIDA
        mov TICKS,#10

;-- 20 SEGUNDOS

        cpl FLAG_20SEGUNDOS

        jb FLAG_20SEGUNDOS,TIMER0_SAIDA

        jb LDR,TIMER0_SAIDA

        clr FLAG_BUZZER

        clr TI
        mov sbuf,#'P'
        jnb TI,$

        setb BUZZER

TIMER0_SAIDA:

        setb TR0                                ; Liga Timer 0

        pop PSW
        pop ACC

        RETI                                    ; Retorna da interrupcao

        END

```


Apêndice 2 – Configuração do VideoCapx

```
##### Software do VídeoCapX #####
Object = "{648A5603-2C6E-101B-82B6-000000000014}#1.1#0"; "MSCOMM32.OCX"
Object = "{831FDD16-0C5C-11D2-A9FC-0000F8754DA1}#2.0#0"; "MSCOMCTL.OCX"
Begin VB.Form Form1
    BorderStyle   = 1 'Fixed Single
    Caption       = "Alarme com video"
    ClientHeight  = 7035
    ClientLeft    = 30
    ClientTop     = 270
    ClientWidth   = 3855
    Icon          = "Form1.frx":0000
    LinkTopic     = "Form1"
    MaxButton     = 0 'False
    ScaleHeight   = 469
    ScaleMode     = 3 'Pixel
    ScaleWidth    = 257
    Begin MSCommLib.MSComm MSComm1
        Left      = 3120
        Top       = 6480
        _ExtentX  = 1005
        _ExtentY  = 1005
        _Version  = 393216
        DTREnable = -1 'True
    End
    Begin VB.CommandButton Command14
        Caption   = "Pausa"
        Enabled   = 0 'False
        Height    = 255
        Left      = 1440
        TabIndex  = 38
        Top       = 6480
        Width     = 1215
    End
    Begin VB.CommandButton Command13
        Caption   = "4"
        BeginProperty Font
            Name     = "Webdings"
            Size     = 9.75
            Charset  = 2
        EndProperty
    End
```

```

    Weight      = 400
    Underline   = 0 'False
    Italic      = 0 'False
    Strikethrough = 0 'False
EndProperty
Height      = 300
Left       = 3420
TabIndex   = 37
Top        = 2040
Width      = 300
End
Begin VB.CommandButton Command1
    Caption   = "..."
    Height    = 300
    Left      = 3132
   TabIndex   = 36
    Top       = 2040
    Width     = 252
End
Begin VB.CheckBox Check3
    Caption    = "Deinterlace"
    ForeColor  = &H80000000&
    Height     = 252
    Left       = 120
   TabIndex    = 35
    Top        = 3120
    Width      = 2892
End
Begin VB.CommandButton Command12
    Caption    = "TV tuner settings"
    Height     = 240
    Left       = 1320
   TabIndex    = 34
    Top        = 600
    Visible    = 0 'False
    Width      = 1812
End
Begin VB.Frame wmvframe
    Caption    = "Video WMV "
    ForeColor  = &H80000000&
    Height     = 1572
    Left       = 120

```

TabIndex = 30
Top = 4080
Width = 3612
Begin VB.ComboBox Combo6

Height = 288
Left = 120
TabIndex = 32
Text = "Combo6"
Top = 480
Width = 3012

End

Begin VB.CommandButton Command11

Caption = "..."
Height = 288
Left = 3120
TabIndex = 31
Top = 480
Width = 252

End

Begin VB.Label Label9

Caption = "WindowsMedia profile:"
ForeColor = &H80000008&
Height = 252
Left = 120
TabIndex = 33
Top = 240
Width = 3012

End

End

Begin VB.CheckBox Check1

Caption = "Captura audio"
ForeColor = &H80000008&
Height = 252
Left = 120
TabIndex = 8
Top = 2760
Width = 2892

End

Begin VB.ComboBox Combo5

Height = 315
Left = 1920
Style = 2 'Dropdown List

```

    TabIndex    = 11
    Top         = 3720
    Width       = 1212
End
Begin VB.ComboBox Combo4
    Height      = 315
    Left        = 120
    Style       = 2 'Dropdown List
    TabIndex    = 10
    Top         = 3720
    Width       = 1812
End
Begin VB.CommandButton Command10
    Caption     = "..."
    Height      = 300
    Left        = 3120
    TabIndex    = 12
    Top         = 3720
    Width       = 252
End
Begin VB.ComboBox Combo3
    Height      = 288
    Left        = 120
    Style       = 2 'Dropdown List
    TabIndex    = 2
    Top         = 840
    Width       = 3012
End
Begin VB.CommandButton Command9
    Caption     = "WebCam"
    Height      = 372
    Left        = 2760
    TabIndex    = 15
    Top         = 6000
    Width       = 972
End
Begin VB.CommandButton Command8
    Caption     = "Formato de áudio"
    Height      = 240
    Left        = 1440
    TabIndex    = 9
    Top         = 3480

```

```

    Width      = 1932
End
Begin VB.CommandButton Command7
    Caption     = "..."
    Height      = 300
    Left        = 1800
    TabIndex    = 4
    Top         = 1440
    Width       = 252
End
Begin VB.CommandButton Command6
    Caption     = "..."
    Height      = 300
    Left        = 3120
    TabIndex    = 1
    Top         = 240
    Width       = 252
End
Begin VB.Timer Timer1
    Interval     = 500
    Left        = 3240
    Top         = 960
End
Begin VB.TextBox Text2
    Height       = 288
    Left        = 2280
    TabIndex    = 5
    Text        = "30"
    Top         = 1440
    Width       = 612
End
Begin VB.CheckBox Check2
    Caption      = "Mostra tempo no video"
    ForeColor    = &H80000000&
    Height       = 252
    Left        = 120
    TabIndex    = 7
    Top         = 2400
    Width       = 2892
End
Begin VB.TextBox Text1
    Height       = 288

```

```

Left      = 120
TabIndex  = 6
Text      = "c:\capture.avi"
Top       = 2040
Width     = 3012
End
Begin VB.ComboBox vidsize
Height    = 288
ItemData  = "Form1.frx":628A
Left      = 120
List      = "Form1.frx":6297
Style     = 2 'Dropdown List
TabIndex  = 3
Top       = 1440
Width     = 1692
End
Begin VB.ComboBox cams
Height    = 288
Left      = 120
Style     = 2 'Dropdown List
TabIndex  = 0
Top       = 240
Width     = 3012
End
Begin VB.CommandButton Command4
Caption    = "Pega uma foto"
Height    = 372
Left      = 120
TabIndex  = 13
Top       = 6000
Width     = 1212
End
Begin VB.CommandButton Command3
Caption    = "Inicia g&ravação"
Height    = 372
Left      = 1440
TabIndex  = 14
Top       = 6000
Width     = 1212
End
Begin VB.Frame aviframe
Caption    = "AVI compression"

```

ForeColor = &H80000008&

Height = 1572

Left = 120

TabIndex = 23

Top = 4080

Width = 3612

Begin VB.ComboBox Combo2

Height = 315

Left = 120

Style = 2 'Dropdown List

TabIndex = 29

Top = 1080

Width = 3012

End

Begin VB.CommandButton Command5

Caption = "..."

Height = 300

Left = 3132

TabIndex = 28

Top = 1080

Width = 252

End

Begin VB.ComboBox Combo1

Height = 315

Left = 120

Style = 2 'Dropdown List

TabIndex = 27

Top = 480

Width = 3012

End

Begin VB.CommandButton Command2

Caption = "..."

Height = 300

Left = 3132

TabIndex = 26

Top = 480

Width = 252

End

Begin VB.Label Label5

Caption = "Video codec:"

ForeColor = &H80000008&

Height = 252

```

    Left      = 120
    TabIndex  = 25
    Top       = 240
    Width     = 3012
End
Begin VB.Label Label6
    Caption    = "Audio codec:"
    ForeColor  = &H80000008&
    Height     = 252
    Left       = 120
    TabIndex   = 24
    Top        = 840
    Width      = 3012
End
End
Begin MSComctlLib.Toolbar Toolbar1
    Align      = 1 'Align Top
    Height     = 630
    Left       = 0
    TabIndex   = 41
    Top        = 0
    Width      = 3855
    _ExtentX   = 6800
    _ExtentY   = 1111
    ButtonWidth = 609
    ButtonHeight = 953
    Appearance = 1
    _Version   = 393216
End
Begin VB.Label Label11
    Height     = 375
    Left       = 240
    TabIndex   = 40
    Top        = 6480
    Width      = 1215
End
Begin VB.Label Label10
    Caption    = "Play:"
    Height     = 255
    Left       = 3360
    TabIndex   = 39
    Top        = 1800

```



```

Width      = 375
End
Begin VB.Label Label8
Caption     = "Fonte de audio:"
ForeColor  = &H80000000&
Height     = 255
Left       = 120
TabIndex   = 22
Top        = 3480
Width      = 1935
End
Begin VB.Label Label7
Caption     = "Entrada:"
ForeColor  = &H80000000&
Height     = 252
Left       = 120
TabIndex   = 21
Top        = 600
Width      = 852
End
Begin VB.Label caplabel
ForeColor  = &H00000000&
Height     = 255
Left       = 120
TabIndex   = 20
Top        = 5640
Width      = 3495
End
Begin VB.Label Label4
Caption     = "Framerate:"
ForeColor  = &H80000000&
Height     = 252
Left       = 2280
TabIndex   = 19
Top        = 1200
Width      = 1452
End
Begin VB.Label Label1
Caption     = "Nome do arq. de captura"
ForeColor  = &H80000000&
Height     = 252
Left       = 120

```

```

    TabIndex    = 18
    Top         = 1800
    Width       = 3252
End
Begin VB.Label Label3
    Caption      = "Tamanho do vídeo"
    ForeColor    = &H800000008&
    Height       = 252
    Left         = 120
    TabIndex     = 17
    Top          = 1200
    Width        = 3252
End
Begin VB.Label Label2
    Caption      = "Camera/placa de captura:"
    ForeColor    = &H800000008&
    Height       = 252
    Left         = 120
    TabIndex     = 16
    Top          = 0
    Width        = 1932
End
End
Attribute VB_Name = "Form1"
Attribute VB_GlobalNameSpace = False
Attribute VB_Creatable = False
Attribute VB_PredeclaredId = True
Attribute VB_Exposed = False
Private Declare Function InitCommonControls Lib "Comctl32.dll" () As Long

Dim do_connect, dropped
Private Declare Function AdjustWindowRect Lib "user32" (lpRect As RECT, ByVal dwStyle As Long,
ByVal bMenu As Long) As Long
Private Type RECT
    Left As Long
    Top As Long
    Right As Long
    Bottom As Long
End Type
Private Const WS_CAPTION = &HC00000      ' WS_BORDER Or WS_DLGFRAME

Private Sub cams_Click()

```

```

If cams.ListIndex <> frmVCX.vcx.VideoDeviceIndex Then
    frmVCX.vcx.Connected = False
    connectcam
End If
End Sub

```

```

Private Sub Check1_Click()
Dim v
If Check1.Value And (frmVCX.vcx.DeviceType <> 2) Then
    v = True
Else
    v = False
End If
Command8.Enabled = v
Command10.Enabled = v
Combo4.Enabled = v
Combo5.Enabled = v
If frmVCX.vcx.CaptureAudio <> Check1.Value Then
'   frmVCX.vcx.Connected = False
'   connectcam
frmVCX.vcx.CaptureAudio = Check1.Value
End If
End Sub

```

```

Private Sub Check2_Click()
frmVCX.vcx.Connected = False
If Check2.Value Then
    frmVCX.vcx.UseVideoFilter = True
    frmVCX.vcx.SetTextOverlay 0, "TIME", 0, 0, "Arial", 10, RGB(0, 0, 0), -1
    frmVCX.vcx.SetTextOverlay 1, "TIME", 1, 1, "Arial", 10, RGB(255, 255, 255), -1
Else
    frmVCX.vcx.SetTextOverlay 0, "", 0, 0, "", 0, 0, 0
    frmVCX.vcx.SetTextOverlay 1, "", 0, 0, "", 0, 0, 0
    frmVCX.vcx.UseVideoFilter = False
End If
connectcam
End Sub

```

```

Private Sub Check3_Click()
frmVCX.vcx.UseDeinterlace = Check3.Value
End Sub

```

```

Private Sub Combo1_Click()
frmVCX.vcx.VideoCodecIndex = Combo1.ListIndex - 1
End Sub

Private Sub Combo2_Click()
frmVCX.vcx.AudioCodecIndex = Combo2.ListIndex - 1
End Sub

Private Sub Combo3_Click()
frmVCX.vcx.VideoInputIndex = Combo3.ListIndex
If Combo3 = "Video tuner" Then Command12.Visible = True: frmVCX.vcx.PreviewAudio = True Else
Command12.Value = False
End Sub

Private Sub Combo4_Click()
frmVCX.vcx.Connected = False
connectcam
End Sub

Private Sub Combo5_Click()
frmVCX.vcx.AudioInputIndex = Combo5.ListIndex
End Sub

Private Sub Combo6_Click()
frmVCX.vcx.ProfileIndex = Combo6.ListIndex
End Sub

Private Sub Command1_Click()
CommonDialog1.Filter = "AVI files (*.avi)|*.avi|All fiels (*.*)|*.*"
CommonDialog1.DefaultExt = ".avi"
CommonDialog1.filename = ""
CommonDialog1.ShowSave
If CommonDialog1.filename <> "" Then
    Text1.Text = CommonDialog1.filename
End If
End Sub

Private Sub Command10_Click()
frmVCX.vcx.ShowAudioSourceDlg
End Sub

Private Sub Command12_Click()

```

frmTV.Show

End Sub

Private Sub Command13_Click()

Dim a As New frmPlay

a.Show

a.vcx.PlayerOpen Text1

a.vcx.PlayerStart

End Sub

Private Sub Command14_Click()

If Command14.Caption = "Pause" Then

frmVCX.vcx.PauseCapture

Command14.Caption = "Resume"

Else

frmVCX.vcx.ResumeCapture

Command14.Caption = "Pause"

End If

End Sub

Private Sub Command2_Click()

'frmVCX.vcx.VideoCodecIndex = Combo1.ListIndex - 1

If Combo1.ListIndex Then frmVCX.vcx.ShowVideoCodecDlg

End Sub

Private Sub Command3_Click()

If frmVCX.vcx.IsCapturing Then

Dim ww As Long, hh As Long, frames As Long, dr As Long, mse As Long, fc As Long

frmVCX.vcx.GetCapStatus ww, hh, frames, dr, mse, fc

caplabel.Caption = frames & " frames (" & dr & " dropped) - " & Format(mse / 1000, "Fixed") & " sec."

frmVCX.vcx.StopCapture

Command3.Caption = "Start &recording"

Form1.Caption = "VideoCapX sample app"

Else

```

Command3.Caption = "Stop &recording"
frmVCX.vcx.CaptureRate = Val(Text2)
frmVCX.vcx.CapFilename = Text1.Text
Form1.Caption = "Capturing " + frmVCX.vcx.CapFilename
frmVCX.vcx.HalfSizedVideo = True
frmVCX.vcx.StartCapture
End If
End Sub

```

```

Private Sub Command4_Click()
'If frmVCX.vcx.UseVideoFilter = vcxNo Then MsgBox "For this function to work, you must have 'show
time on video' enabled"
Form2.Picture = frmVCX.vcx.GrabFrame
Form2.Show
Dim r As RECT
r.Left = 0
r.Top = 0
r.Bottom = frmVCX.vcx.VideoHeight
r.Right = frmVCX.vcx.VideoWidth
AdjustWindowRect r, WS_CAPTION, 1

```

```

Form2.Width = Form2.ScaleX(r.Right - r.Left, 3, 1)
Form2.Height = Form2.ScaleY(r.Bottom - r.Top, 3, 1)
End Sub
Private Sub Command5_Click()
frmVCX.vcx.AudioCodecIndex = Combo2.ListIndex - 1
If Combo2.ListIndex Then frmVCX.vcx.ShowAudioCodecDlg
End Sub

```

```

Private Sub Command6_Click()
frmVCX.vcx.ShowVideoSourceDlg
End Sub

```

```

Private Sub Command7_Click()
frmVCX.vcx.ShowVideoFormatDlg
End Sub

```

```

Private Sub Command8_Click()
frmAFmt.Show
End Sub

```

```

Private Sub Command9_Click()

```

frmUpload.Show

End Sub

Function file2str(filename) As String

Dim X

Open filename For Input As #1

While Not EOF(1)

X = X + Input(1, #1)

Wend

Close #1

file2str = X

End Function

Private Sub Command11_Click()

CommonDialog1.Filter = "PRX files (*.prx)|*.prx|All fiels (*.*)|*.*"

CommonDialog1.DefaultExt = ".prx"

CommonDialog1.filename = ""

CommonDialog1.ShowOpen

If CommonDialog1.filename <> "" Then

Dim X As String

X = file2str(CommonDialog1.filename)

frmVCX.vcx.ProfileData = X

Combo6.ListIndex = -1

Combo6.Text = frmVCX.vcx.GetProfileName(-1)

End If

End Sub

Private Sub Form_Activate()

Text1_Change

End Sub

Private Sub Form_Initialize()

InitCommonControls

End Sub

Private Sub Form_Load()

' Buffer to hold input string

Dim Instring As String

' Use COM1.

MSComm1.CommPort = 1

MSComm1.RThreshold = 1

```
' 9600 baud, no parity, 8 data, and 1 stop bit.  
MSComm1.Settings = "9600,N,8,1"  
' Open the port.  
MSComm1.PortOpen = True  
' Send the attention command to the modem.
```

```
'frmVCX.vcx.PreviewAudio = True  
do_connect = False  
'frmVCX.vcx.UserFilterCLSID = "{5BBB3392-A56A-4EFF-A418-100BF62B1375}"  
frmVCX.vcx.DebugMode = 1  
'frmVCX.vcx.UseVMR9 = True  
frmVCX.Show  
Me.Move Val(GetSetting("VideoCapXapp", "settings", "leftpos", "300")),  
Val(GetSetting("VideoCapXapp", "settings", "toppos", "300"))  
frmVCX.Move Me.Left + Me.Width, Me.Top  
If frmVCX.vcx.GetVideoDeviceCount = 0 Then MsgBox "no video hardware found!": End  
For f = 0 To frmVCX.vcx.GetVideoDeviceCount - 1  
    cams.AddItem frmVCX.vcx.GetVideoDeviceName(f)  
Next f  
  
Combo6.Clear  
On Error Resume Next  
For f = 0 To frmVCX.vcx.GetProfileCount - 1  
    Combo6.AddItem frmVCX.vcx.GetProfileName(f)  
Next f  
If Combo6.ListCount Then Combo6.ListIndex = 0: frmVCX.vcx.ProfileIndex = Combo6.ListIndex  
  
Combo4.Clear  
For f = 0 To frmVCX.vcx.GetAudioDeviceCount - 1  
    Combo4.AddItem frmVCX.vcx.GetAudioDeviceName(f)  
Next f  
Combo1.AddItem "(none)"  
For f = 0 To frmVCX.vcx.GetVideoCodecCount - 1  
    Combo1.AddItem frmVCX.vcx.GetVideoCodecName(f)  
Next f  
Combo2.AddItem "(none)"  
For f = 0 To frmVCX.vcx.GetAudioCodecCount - 1  
    Combo2.AddItem frmVCX.vcx.GetAudioCodecName(f)  
Next f
```


On Error Resume Next

```
'cams.ListIndex = Val(GetSetting("VideoCapXapp", "settings", "1"))  
'vidsize.ListIndex = Val(GetSetting("VideoCapXapp", "settings", "2"))  
Text1.Text = GetSetting("VideoCapXapp", "settings", "3", "c:\capture.avi")  
Text2.Text = GetSetting("VideoCapXapp", "settings", "4", "30")  
Combo1.ListIndex = Val(GetSetting("VideoCapXapp", "settings", "5", "-1"))  
Combo2.ListIndex = Val(GetSetting("VideoCapXapp", "settings", "6", "-1"))  
'Combo3.ListIndex = Val(GetSetting("VideoCapXapp", "settings", "63", "-1"))  
'Combo4.ListIndex = Val(GetSetting("VideoCapXapp", "settings", "7", "1"))  
Check2.Value = Val(GetSetting("VideoCapXapp", "settings", "8"))  
Check1.Value = Val(GetSetting("VideoCapXapp", "settings", "ca"))
```

Check1_Click

```
cams.ListIndex = 0  
If Combo4.ListCount Then Combo4.ListIndex = 0  
do_connect = True  
connectcam  
On Error Resume Next  
Combo5.ListIndex = Val(GetSetting("VideoCapXapp", "settings", "9"))  
End Sub
```

Private Sub Form_Unload(Cancel As Integer)

```
If frmVCX.vcx.IsCapturing Then frmVCX.vcx.StopCapture  
frmVCX.vcx.Connected = False
```

```
SaveSetting "VideoCapXapp", "settings", "1", Str(cams.ListIndex)  
SaveSetting "VideoCapXapp", "settings", "2", Str(vidsize.ListIndex)  
SaveSetting "VideoCapXapp", "settings", "3", Text1.Text  
SaveSetting "VideoCapXapp", "settings", "4", Text2.Text  
SaveSetting "VideoCapXapp", "settings", "5", Combo1.ListIndex  
SaveSetting "VideoCapXapp", "settings", "6", Combo2.ListIndex  
SaveSetting "VideoCapXapp", "settings", "63", Combo3.ListIndex  
SaveSetting "VideoCapXapp", "settings", "7", Combo4.ListIndex  
SaveSetting "VideoCapXapp", "settings", "8", If(Check2.Value, "1", "0")  
SaveSetting "VideoCapXapp", "settings", "ca", If(Check1.Value, "1", "0")  
SaveSetting "VideoCapXapp", "settings", "9", Combo5.ListIndex  
SaveSetting "VideoCapXapp", "settings", "leftpos", Form1.Left  
SaveSetting "VideoCapXapp", "settings", "toppos", Form1.Top  
MSComm1.PortOpen = False
```

End

End Sub

Private Sub MSComm1_OnComm()

If MSComm1.CommEvent = comEvReceive Then

Dim sData As String

sData = MSComm1.Input

Código inserido para Gravação do Vídeo

If sData = "A" Then (SINAL EMITIDO PELA PORTA SERIAL)

Command3.Caption = "Stop &recording"

'frmVCX.vcx.CaptureRate = Val(Text2)

frmVCX.vcx.CapFilename = Text1.Text

Form1.Caption = "Capturing " + frmVCX.vcx.CapFilename

'frmVCX.vcx.HalfSizedVideo = True

frmVCX.vcx.StartCapture

Label11.Caption = "GRAVANDO..."

Else

Dim ww As Long, hh As Long, frames As Long, dr As Long, mse As Long, fc As Long

frmVCX.vcx.GetCapStatus ww, hh, frames, dr, mse, fc

caplabel.Caption = frames & " frames (" & dr & " dropped) - " & Format(mse / 1000, "Fixed") & " sec."

frmVCX.vcx.StopCapture

Command3.Caption = "Start &recording"

Form1.Caption = "VideoCapX sample app"

Label11.Caption = "GRAVAÇÃO ENCERRADA!"

End If

End If (TÉRMINO DO SINAL)

Término do Código

End Sub

Private Sub Text1_Change()

```

Dim t
t = Text1.Text
If UCase(Right(t, 4)) = ".WMV" Then aviframe.Visible = False Else aviframe.Visible = True
wmvframe.Visible = Not aviframe.Visible
End Sub

Private Sub Text1_Validate(Cancel As Boolean)
frmVCX.vcx.CapFilename = Text1
End Sub

Private Sub Text2_Validate(Cancel As Boolean)
    frmVCX.vcx.Connected = False
    connectcam 'frmVCX.vcx.CaptureRate = Val(Text2)
End Sub

Private Sub Timer1_Timer()
Dim ww As Long, hh As Long, frames As Long, dr As Long, mse As Long, fc As Long
If frmVCX.vcx.IsCapturing Then
    frmVCX.vcx.GetCapStatus ww, hh, frames, dr, mse, fc
    caplabel.Caption = frames & " frames (" & dr & " dropped) - " & Format(mse / 1000, "Fixed") & "
sec."
' If dr - dropped > 0 Then frmVCX.vcx.AddBlankFrames dr - dropped: dropped = dropped + dr
    Command14.Enabled = True
Else
    Command14.Enabled = False
End If
Dim i
i = frmVCX.vcx.GetActualFrameRate
Label4.Caption = "Framerate:" & Format(i, "0.00")
End Sub

Private Sub vcx_CaptureEnd()
Timer1.Interval = 0
Form1.Caption = "VideoCapX sample app"
Command3.Caption = "Start &recording"
End Sub

Private Sub vcx_CaptureStart()
Timer1.Interval = 500
End Sub

Private Sub vidsize_Click()

```

```

'set by width x height
Dim i
i = InStr(vidsize.Text, "x")
frmVCX.vcx.SetVideoFormat Val(vidsize.Text), Val(Mid(vidsize.Text, i + 1))
X = Right(vidsize.Text, 2)
If Left(X, 1) = "," Then X = Mid(X, 2)
frmVCX.vcx.ColorFormat = X

```

```

Dim r As RECT
r.Left = 0
r.Top = 0
r.Bottom = frmVCX.vcx.VideoHeight
r.Right = frmVCX.vcx.VideoWidth
AdjustWindowRect r, WS_CAPTION, 0
frmVCX.Width = frmVCX.ScaleX(r.Right - r.Left, 3, 1)
frmVCX.Height = frmVCX.ScaleY(r.Bottom - r.Top, 3, 1)
End Sub

```

```

Sub refreshvidsize()
Dim a, f
a = frmVCX.vcx.GetVideoCaps()
If VarType(a) Then
    If UBound(a) >= LBound(a) Then
        vidsize.Clear
        For f = LBound(a) To UBound(a)
            vidsize.AddItem a(f, 0) & "x" & a(f, 1) & "x" & a(f, 2) & "," & a(f, 3)
        Next f
    End If
End If
vidsize.Text = vidsize.List(0)
End Sub

```

```

Sub refreshvideoinput()
Dim f
Combo3.Clear
For f = 0 To frmVCX.vcx.GetVideoInputCount - 1
    Combo3.AddItem frmVCX.vcx.GetVideoInputName(f), f
Next f
If frmVCX.vcx.GetVideoInputCount Then Combo3.ListIndex = 0
End Sub

```

```

Sub connectcam()
If Not do_connect Then Exit Sub

```

```

frmVCX.vcx.VideoDeviceIndex = cams.ListIndex
frmVCX.vcx.AudioDeviceIndex = Combo4.ListIndex
frmVCX.vcx.CaptureAudio = Check1.Value
frmVCX.vcx.CaptureRate = Val(Text2)
'frmVCX.vcx.SetCrop 50, 50, 200, 100
frmVCX.vcx.UseVideoFilter = Check2.Value
frmVCX.vcx.Connected = True
refreshvidsize
frmVCX.vcx.ColorFormat = 0
updatecombo5
refreshvideoinput
frmVCX.vcx.Preview = True
If Check2.Value Then
    frmVCX.vcx.SetTextOverlay 0, "TIME", 0, 0, "Arial", 10, RGB(0, 0, 0), -1
    frmVCX.vcx.SetTextOverlay 1, "TIME", 1, 1, "Arial", 10, RGB(255, 255, 255), -1
Else
    frmVCX.vcx.SetTextOverlay 0, "", 0, 0, "", 0, 0, 0
    frmVCX.vcx.SetTextOverlay 1, "", 0, 0, "", 0, 0, 0
End If
frmVCX.vcx.VideoCodecIndex = Combo1.ListIndex - 1
frmVCX.vcx.VideoCodecQuality = 95
frmVCX.vcx.AudioCodecIndex = Combo2.ListIndex - 1
If frmVCX.vcx.DeviceType = 1 Then Command12.Visible = True Else Command12.Visible = False
On Error Resume Next
If frmVCX.vcx.DeviceType = 1 Then Combo3 = "Video tuner"
End Sub

Sub updatecombo5()
Combo5.Clear
For f = 0 To frmVCX.vcx.GetAudioInputCount - 1
    Combo5.AddItem frmVCX.vcx.GetAudioInputName(f), f
Next f
End Sub

```